



Citation for published version:

Cooper, K 2006, *Implementation of a prototype peer learning mathematics tutor for children: A user-centred approach*. Computer Science Technical Reports, no. CSBU-2006-17, Department of Computer Science, University of Bath.

Publication date:
2006

[Link to publication](#)

©The Author December 2006

University of Bath

Alternative formats

If you require this document in an alternative format, please contact:
openaccess@bath.ac.uk

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Department of
Computer Science



UNIVERSITY OF
BATH

Technical Report

Undergraduate Dissertation: Implementation of a Prototype Peer Learning Mathematics Tutor for Children: A User-Centred Approach

Katherine Cooper

Copyright ©December 2006 by the authors.

Contact Address:

Department of Computer Science
University of Bath
Bath, BA2 7AY
United Kingdom
URL: <http://www.cs.bath.ac.uk>

ISSN 1740-9497

**Implementation of a Prototype
Peer Learning Mathematics Tutor for Children:
A User-Centred Approach**

Katherine Cooper
BSc (Hons) Computer Information Systems
2006

Implementation of a Prototype Peer Learning Mathematics Tutor for Children: A User-Centred Approach

Submitted by Katherine Cooper

COPYRIGHT

Attention is drawn to the fact that copyright of this dissertation rests with its author. The Intellectual Property Rights of the products produced as part of the project belong to the University of Bath (see <http://www.bath.ac.uk/ordinances/#intelprop>).

This copy of the dissertation has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no quotation from the dissertation and no information derived from it may be published without the prior written consent of the author.

Declaration

This dissertation is submitted to the University of Bath in accordance with the requirements of the degree of Bachelor of Science in the Department of Computer Science. No portion of the work in this dissertation has been submitted in support of an application for any other degree or qualification of this or any other university or institution of learning. Except where specifically acknowledged, it is the work of the author.

Signed:

This dissertation may be made available for consultation within the University Library and may be photocopied or lent to other libraries for the purposes of consultation.

Signed:

Abstract

Mathematics is a high educational priority, but many children do not reach the target level of attainment by the end of primary school. Mathematics can be difficult to learn and hard to teach, particularly as teachers have large mixed-ability classes. It is proposed that additional support could be provided by a computer based mathematics tutor. However, primary school children rarely work individually at computers, so a software tutor should be suitable for peer learning.

To establish requirements for the proposed solution, relevant literature is researched; existing solutions are evaluated and stakeholders are involved through observations and interviews. A user-centered design approach is adopted; utilising participatory methods with year 5 children. From this, a low-fidelity prototype for a 'Maths Zoo' game emerges. The design is formalised and implemented into a high-fidelity prototype. Two phases of formative stakeholder evaluations influence the prototype's re-design and improve its usability.

The final prototype is suitable for interactive whiteboards; it features zoo related division questions, each with a 'help activity' that brings the problem to a concrete level using visual representations that can be directly manipulated. User experience evaluations indicate that children in year 5 think the 'Maths Zoo' is a fun way to learn. Although learning evaluations with two year 5 classes do not allow firm conclusions to be made, it is suggested that the prototype is better suited to low-ability children.

Enhancements to the 'Maths Zoo' are suggested and a framework for participatory design with children is proposed.

Acknowledgements

I would like to thank Hilary Johnson for her advice and guidance throughout the project. I would also like to thank the teachers at Moorlands Junior School for their involvement, especially Steve Bailey. Further thanks go to the children in year 5 at Moorlands Junior School, particularly those in the design team for their enthusiasm and creative ideas. Thanks also go to the sisters involved in the pilot sessions and to Alice Morrissey for her endearing animal drawings.

Contents

1	INTRODUCTION	1
1.1	Aims.....	2
1.2	Objectives	3
2	LITERATURE SURVEY	4
2.1	Introduction.....	4
2.2	The Mathematics National Curriculum at Key Stage Two	4
2.3	Children's Cognitive Development and Learning	5
2.4	Groupware and Collaborative Technology	9
2.5	Usability Guidelines.....	12
2.6	Understanding Users and Interaction Styles	14
2.7	Involving Users.....	15
2.8	Conclusion	20
3	REQUIREMENTS ANALYSIS AND SPECIFICATION	22
3.1	Introduction.....	22
3.2	Requirements Analysis	22
3.3	Requirements Specification	29
3.4	Conclusion	37
4	DESIGN METHODS.....	38
4.1	Introduction.....	38
4.2	Participatory Design.....	38
4.3	Prototyping	45
4.4	Conclusion	45
5	USER INTERFACE AND SYSTEM DESIGN	47
5.1	Introduction.....	47
5.2	High-Level Design Decisions	47
5.3	System Architecture.....	48
5.4	Detailed Design.....	50
5.5	Conclusion	57
6	USER-CENTRED RE-DESIGNS	58
6.1	Introduction.....	58
6.2	User-Centred Re-Design: Phase One	58
6.3	User-Centred Re-Design: Phase Two	63
6.4	Conclusion	70
7	IMPLEMENTATION AND TESTING	71
7.1	Introduction.....	71
7.2	Implementation Language Choice	71
7.3	About Flash.....	72
7.4	Prototype Implementation.....	73
7.5	Testing	82
7.6	Conclusion	82

8	EVALUATION	83
8.1	Introduction.....	83
8.2	Requirements Verification	83
8.3	Learning Evaluation.....	84
8.4	User Experience Evaluation.....	90
8.5	Conclusion	91
9	CONCLUSION	92
9.1	The Future	97
10	BIBLIOGRAPHY.....	99
11	APPENDICES.....	103
A	Eight Golden Rules of Interface Design	104
B	Usability Heuristics.....	105
C	The Mechanics of Collaboration.....	105
D	Education City Screen Shots.....	106
E	Primary Games Screen Shots.....	109
F	Pilot Brainstorming Session Spider Diagrams	110
G	Brainstorming Session Spider Diagrams	111
H	CARD Session Ordered Cards.....	112
I	PICTIVE Session Low-Fidelity Prototype.....	113
J	Initial Game Instructions Text	114
K	Maths Zoo Alpha Prototype.....	115
L	Maths Zoo Beta Prototype	122
M	Maths Zoo 1.0 Prototype.....	131
N	Full Listing of Maths Zoo Questions	138
O	Test Case Sample	139
P	Requirements Verification	141
Q	Learning Evaluation Written Division Tests.....	146
R	Learning Evaluation Scripts.....	147
S	Learning Evaluation Results: Class 1	148
T	Learning Evaluation Results: Class 2	150
U	User Experience Questionnaire Results	153
V	Source Code.....	154

1 Introduction

Mathematics is a fundamental part of education because it provides the means to understand and deal with our physical environment. For this reason, it is a core and statutory subject in schools, with primary school children receiving mathematics lessons daily. However, research shows that the proportion of children reaching the expected level of attainment by the end of primary school is significantly below target (OFSTED 2005). The reasoning for this could be linked to the notion that mathematics is 'difficult to learn and hard to teach' (Wood 1998, p.227).

Essentially, mathematics involves the discovery and application of numerical and spatial relationships (Hopkins et al. 1999), which is underpinned by the ability to form patterns, generalizations and abstractions in different contexts. However, it is this abstract nature that can cause difficulties for children when learning mathematics. Piaget's theories of cognitive development imply that the thinking of children between the ages of seven and 11, (which corresponds to key stage two), is linked to physical reality. Furthermore, Piaget believed that children are unable to solve abstract problems until they have reached the next stage of development, which usually occurs from age 11 onwards.

Problems with mathematical learning can also be linked to children forming misconceptions or 'inventing' rules to explain patterns they see. Clearly, teaching needs to be carried out so that these misconceptions can be exposed and learned from. However, with large class sizes, teachers are faced with challenges because they cannot necessarily give one-on-one tuition to every child. Also, mathematical abilities in classes are likely to vary and teachers should attempt to provide content and methods that are suitably challenging in view of the child's ability level. Again, this is difficult where there are so many children to consider.

Furthermore, there can be problems related to interest, motivation and confidence with children and mathematics, which can cause barriers to learning. Therefore it is important to make mathematics interesting so children are motivated to persist. Research indicates that there is a moderate correlation between confidence and achievement (Askew and Wiliam 1995); therefore this should be capitalized on through teaching that builds confidence.

Clearly, there is scope for improvement and Information and Communication Technology (ICT) may well provide a complementary means of mathematical tuition. As stated by Askew and Wiliam (1995, pp.34-35), 'computers can have a substantial positive impact on pupils achievement in maths'. It has also been found by OFSTED (2004) that attitudes and responses to ICT in lessons by pupils are positive. However, OFSTED (2005) have also found that 'ICT is underused to support teaching and learning in mathematics', thus indicating a need to further increase ICT usage.

It is necessary to consider what a computer based solution could provide to improve a child's learning of mathematics. Firstly, a software based tutor could allow abstract concepts to be visualised in concrete ways, for example, through use of graphical metaphors. With metaphors, prior knowledge is used to understand new situations. Also, it may be possible to do things that would not be feasible in a classroom. This opens up potential for mathematics to become more interesting by using creative design concepts. Furthermore, it may be possible to tailor the content and feedback to that of the learner's individual needs.

This potential for individual, tailored tutoring is important; however, children in primary schools rarely work individually at computers due to computer availability (OFSTED 2004). More often than not, children will work in pairs and with this in mind, peer learning and collaboration should be considered because software for schools is unlikely to be used individually. This is not necessarily a bad thing, as there are many benefits from peer learning. Children can learn from each other and discuss concepts in familiar terms.

However, research (Monteith 1998) has shown that computer use does not necessarily lead to effective collaboration, one reason being that children often spend a lot of time trying to make the use of the computer 'fair'. In view of this, perhaps collaboration should be explicitly supported so that this is not an issue. The use of interactive whiteboards, now commonly found in classrooms, may well provide this explicit support. Alternatively, software should be designed so that the focus is on working together. Clearly there are challenges in designing such software.

A further area of concern is that of making software usable and appealing for the intended user group in view of their capabilities and interests. Research is obviously needed into how children develop and learn; as well as research around how to design usable systems by considering usability guidelines. Additionally, consultation with teachers is needed as they have practical experience from working with children that is likely to lead to a good understanding of the user group. However, this cannot be relied on alone; it is easy to make assumptions and form misconceptions about the users if they are not consulted.

It is, therefore, important to consult and involve children in the design of software intended for their use. Where a user-centred approach is taken, it is common to use participatory design so that the users become equal partners in the design team and the product is designed in co-operation with the designers (Preece et al. 2002). However, according to Druin (1999), the approach needs to be adapted and changed to suit the children. There is also some concern over what the children will be able to contribute, which is linked to their ability to articulate their needs and them knowing what is best for them in a learning context.

With a user-centred approach, user involvement should continue throughout the design and development process. Through iterative prototype evaluations with users, it is possible to refine the design, thus the resulting system is likely to be more usable. Furthermore, it is necessary to investigate whether a system will actually make a difference in mathematical learning. To do this, carefully planned learning evaluations should be carried out.

Based on this discussion around the problem domain, the aims and objectives will now be identified:

1.1 Aims

To design and implement a software prototype of a peer learning mathematics tutoring for key stage two children. This should be achieved by considering relevant literature in the problem domain and by adopting an iterative user-centred design approach that utilises participatory design and user evaluations.

1.2 Objectives

- To research and review literature relevant to the problem domain, including:
 - Topics related to learning and mathematics, such as the mathematics national curriculum; cognitive development; motivation and peer learning
 - Topics related to technology and usability, including collaboration and groupware; usability guideline and interaction styles
 - Topics related to understanding and involving users, including ethnography and participatory design amongst other methods
- To carry out research in a primary school by means of lesson observations and teacher interviews with a specific focus on finding out about key stage two mathematics teaching and the use of computers and technology in schools, including interactive whiteboards
- To carry out interviews with children in key stage two to find out their opinions about mathematics and computers, as well as investigating their interests
- To carry out evaluations of relevant existing software, such that positive design aspects can be identified and carried forward and negative aspects specifically excluded from the design of a peer learning mathematics tutor
- To establish a set of requirements based on the researched literature, stakeholder involvement and existing system evaluations
- To involve a group of children in key stage two in the design of a peer learning mathematics tutor, by use of participatory design adapted for use with children
- To implement a prototype interactive peer learning mathematics tutor for key stage two children based on relevant literature, requirements and primarily the user-centred design
- To carry out evaluations with children in key stage two to assess the usability of the system in order to influence re-design
- To carry out evaluations with teachers of key stage two children to assess the appropriateness and usability of the system to influence re-design
- To carry out learning evaluations with children in key stage two to assess the value that the implemented prototype can add to their learning of mathematics
- To draw conclusions relating to the reviewed literature, the processes involved during the project, the usability of the system and educational value of the system
- To suggest improvements and areas of further work based on the project findings

It is now necessary to tackle each of these objectives. Firstly, relevant literature will be surveyed in the next chapter to give a theoretical grounding to the project. Following this, the project will move towards establishing requirements before the iterative user-centred approach to design is detailed. The resulting design and prototype implementation will then be discussed and summative evaluations carried out. Finally, conclusions will be made about the project as a whole and future work identified.

2 Literature Survey

2.1 Introduction

It is necessary to research, critique and draw together relevant areas in the problem domain to set a firm grounding to the project. Firstly, mathematics in key stage two will be investigated as to identify expectations of children's mathematical knowledge and understanding in this period of schooling. Following this, in order to influence the content and functionality of the peer learning mathematics tutor, consideration will be made of children's learning and cognitive development, particularly in terms of mathematical and peer learning.

Subsequently, different types of collaborative technology, including interactive whiteboards, will be discussed to determine their potential for use in a peer learning context. Furthermore, Human-Computer Interaction (HCI) guidelines and interaction styles should be investigated to provide a framework towards developing a usable system. Lastly, user-centred design approaches will be explored. From this it should be possible to shape the methods and activities used so that the needs, capabilities and interests of the users are appropriately catered for.

2.2 The Mathematics National Curriculum at Key Stage Two

The national curriculum defines the content and targets of mathematics teaching in schools. The curriculum is organized into four key stages, with key stage two corresponding to the primary school years of 3 to 6 where children are aged between seven and 11. Within the mathematics national curriculum are three main areas (see National Curriculum Online 2005b for further details):

- **Number and algebra** - Pupils use the number system more confidently. They move from counting reliably to calculating fluently with all four number operations
- **Shape, Space and Measures** - Pupils explore features of shape and space and develop their measuring skills in a range of contexts
- **Handling Data** - Pupils discuss and present their methods and reasoning using a wider range of mathematical language, diagrams and charts

A key goal in primary education, as set in the Primary National Strategy (DfES 2003), is to 'combine excellence in teaching with enjoyment of learning'. Clearly if a software based mathematics tutor is to be appropriate for use by children in key stage two, it should provide enjoyment whilst allowing them to develop and learn in areas as defined in the curriculum. Achieving this is likely to be a challenge, particularly in facilitating learning. However, the strategy has also detailed principles for good learning and teaching which could be used to influence the design of the system. These principles include:

- **Building on what learners already know** - Structuring and pacing teaching so that students know what is to be learnt, how and why
- **Making learning vivid and real** - Developing understanding through enquiry, creativity, e-learning and group problem solving
- **Making learning an enjoyable and challenging experience** - stimulating learning through matching teaching techniques and strategies to a range of learning styles

Another pertinent area in this Primary National Strategy is the notion that ‘Learning must be focused on individual pupils’ needs and abilities’. In the key stages, the abilities of children are formally categorised by levels. These levels define the ‘types and range of performance that pupils working at that level should characteristically demonstrate’ (National Curriculum Online 2005a). Within any class there will be a range of mathematical abilities and it is not always appropriate to use the same methods and examples for every child. Therefore, it is important to understand a child’s ability level so that tuition can be tailored to their needs.

2.3 Children’s Cognitive Development and Learning

A child’s ability level is clearly linked to their cognitive development and it is necessary to understand a child’s level of cognitive development in order to adjust the methods and examples used. A child must be taught at the right level; if a task is too easy, the child may get bored; conversely, if the task is too difficult they may be discouraged. This is known as *the problem of the match* (Hunt 1961, cited by Woolfolk 2004). To understand these levels of development further, different theories of cognitive development will be discussed.

Development related to changes in thinking is known as cognitive development. Cognitive development is the ‘gradual orderly changes by which mental processes become more complex and sophisticated’ (Woolfolk 2004, p.24). Cognitive development is responsible for the ‘acquisition and development of knowledge and cognitive processes such as language, memory, problem-solving and drawing’ (Lee and Das Gupta 1995, p.3).

Key theories of cognitive development that have influenced teaching and learning in schools include those by Piaget and Vygotsky. These theories also have implications for software that is intended to aid mathematical learning. They can be used to guide the type of content and interaction used in the system based on the expected capabilities of the user.

2.3.1 Piaget’s Cognitive Development Theories

The Swiss psychologist Jean Piaget theorised about cognitive development from infancy to adulthood. His theories were constructivist, in that they ‘assume the active building-up of knowledge and cognitive processes from very simple starting points’ (Das Gupta 1994, cited by Lee and Das Gupta 1995, p.5). Piaget claimed that children take the long journey to adult knowledge and thinking through stages that are qualitatively different from each other. He theorised about four stages of cognitive development that children pass through in order and generally at specific ages. These are:

- 1. Sensorimotor (birth to 2 years)** - An infant’s thinking and behaviour is limited to physical actions that can be performed with objects in their environment.
- 2. Pre-operational (2 to 7 years)** - A child moves towards operational thinking, whereby actions are carried out or reversed mentally rather than physically, through the use of symbolic representation. However, their thinking is rigid; they cannot deal with several aspects of a situation at one time. Their thought is limited to one direction, with a lack of reversibility. A child also has semilogical reasoning; meaning their thoughts are loosely linked rather than in logical relationships. Children in this phase tend to be ego-centric and assume others experience the world as they do.

3. **Concrete operational (7 to 11 years)** - A child develops a logical system of thinking linked to physical reality, but cannot yet reason about hypothetical, abstract problems with several factors. They realize objects can be changed yet retain some of their original characteristics and they understand some changes can be reversed. They understand identity; so realise objects and people remain the same over time. They grasp the concept of compensation, so realise changes in one dimension can be offset by changes in another dimension. They develop the ability to perform reversible thinking. Their understanding of identity, compensation and reversibility allows them to solve conservation problems. A child also develops classification skills, allowing them to group objects into categories, however, they understand that objects can fit into more than one classification. They can also order objects sequentially according to one aspect.
4. **Formal operational (11 to adult)** - A child/adult develops abstract thinking. They can demonstrate hypothetico-deductive reasoning which is a problem solving strategy for hypothetical situations. However, it is possible to remain in the concrete-operational stage unless problems are experienced that cannot be solved using concrete-operations.

Piaget believed children move from one stage to another through the transitional processes of assimilation, accommodation and equilibrium. These processes are driven by peoples' natural tendencies towards organization, which is the 'ongoing process of arranging information and experience into mental systems or categories', known as schemes (Woolfolk 2004, p.30).

The assimilation process tries to fit new information within existing schemes as best as possible, however, sometimes the information must be distorted to fit. With accommodation, existing schemes are either changed in response to the new information or new schemes are developed if existing schemes are not appropriate. Equilibration is the 'keystone of developmental change within Piaget's system' (Lee and Das Gupta 1995, p.34). Piaget suggested that when children are satisfied with their mode of thought, they are in equilibrium. However, when they are aware of shortcomings in their existing thinking, they become dissatisfied and therefore, are in disequilibrium. To correct this, they adopt more sophisticated ways of thinking to reach a more stable equilibrium (Lee and Das Gupta 1995).

In view of Piaget's theories, it would be expected that children in key stage two will be in the concrete operational stage, meaning that their thinking is linked to concrete objects. This should be considered when designing software for this age group, as the content and interaction with the system may need to use concrete representations of the abstract concepts found in mathematics. One way of achieving this could be through use of appropriate metaphors, which as described by Preece et al. (1994), allow abstract concepts to be conveyed in a more familiar and accessible ways. Furthermore, it is thought that by drawing on prior knowledge, as occurs with assimilation, a learner can more readily develop an understanding of the new domain, yet they should be stretched sufficiently to cause disequilibrium.

A further implication of this research relates to the involvement of children in designing software. Software can be considered as abstract, thus potentially making it difficult for children in Piaget's concrete operational stage of cognitive development to contribute to its design. However, by making the design process concrete, for example, through use of pens, paper and other tangible objects it should be possible to overcome such problems.

2.3.2 Vygotsky's Cognitive Development Theories

Although Piaget's theories are considered comprehensive; certain aspects are disputed by other theorists. Piaget argued that cognitive structures develop without any direct teaching from adults, however, the Russian psychologist, Lev Vygotsky disagreed. Vygotsky considered the impact of cultural context on development. He believed interactions with adults and peers are essential for cognitive development. Additionally, he thought instruction is essential to reach the highest levels of thinking.

Vygotsky believed functions must be experienced at a social level before children are able to internalise them. 'Higher mental processes appear first between people as they are co-constructed during shared activities' (Woolfolk 2004, p.45). Later, children may be able to function independently to solve similar problems. Vygotsky suggested that children's cognitive development is fostered by interactions with people who are more capable or advanced in their thinking.

He believed that children have a *zone of proximal development*, this is the range in which a child is able to successfully solve a problem with help from an adult or more experienced peer. This can be applied to teaching, by putting students in situations 'where they have to reach to understand, but where support from other students or from the teacher is also available' (Woolfolk 2004, p.52). This zone of proximal development should be considered when planning the support provided when children learn and problem solve. Adult assistance, known as *scaffolding* can be used by children as 'support while they build a firm understanding that will eventually allow them to solve the problems on their own' (Wood, Bruner and Ross 1976, cited by Woolfolk 2004, p.50).

In view of Vygotsky's ideas, a software tutor could be used to provide instruction, as this is seen as essential in facilitating cognitive development. Furthermore, a software tutor could be used to provide suitably challenging tasks such that the child is stretched within this zone of proximal development, whilst providing scaffolding as they learn. Scaffolding is generally provided through assistance from a more experienced person, be it a peer or adult. However, with a software tutor, this assistance and help could be replicated, possibly through use of an interface persona to which the child can relate to.

Further to this consideration of cognitive development, it is also necessary to understand how children learn and acquire knowledge, with a view to supporting and aiding these processes in a software based tutor. Additionally, the link between reinforcement and retention should be investigated as this is likely to have implications for such a system. Furthermore, the concept of motivation and its relationship to learning should be looked into because the system may need to attempt to motivate if positive effects on learning can be expected.

2.3.3 Knowledge, Learning, Reinforcement and Motivation

Learning is the process through which experience causes permanent change in knowledge or behaviour (Woolfolk 2004). Knowledge can be considered the outcome of learning, however, knowledge can be important in understanding and remembering new information (Recht and Leslie 1988, cited by Woolfolk 2004).

In order to retain information permanently, information must be moved from working memory into long term memory. This is generally achieved through repetition or connecting new information to existing knowledge, as with assimilation, however this requires time and effort. Again, this further enforces the need to relate mathematical learning to concrete objects that children are familiar with.

The way information is learnt can strongly affect later recall. Some psychologists believe that information coded both visually and verbally is easier to learn (Mayer and Sims 1994, cited by Woolfolk 2004). From this it can be seen that benefits could be derived from using a computer based tutor because it could provide the ability to represent information visually through use of a Graphical User Interface (GUI); and verbally through audio output if the appropriate technology is available.

According to Woolfolk (2004), for learning to be effective, children need to pay and maintain attention to a stimulus. Teaching should be meaningful, through use of vocabulary that makes sense to the children. Furthermore, learning should be through distributed practice, meaning learning is in short periods with intervals for rest as this is more likely to lead to long-term memory formation. Moreover, children need opportunities for practice and they need to receive feedback that guides them away from making the same mistakes again as 'studies have concluded that it is more helpful to tell students why they are wrong so they can learn more appropriate strategies' (Woolfolk 2004, p.565). In view of this, a software tutor should seek to follow these strategies to make learning effective.

Woolfolk (2004) also discusses using various types of reinforcement at different stages of learning. When learning a new behaviour, individuals learn faster with continuous reinforcement. Behaviours are best maintained by an intermittent schedule of reinforcement. Persistence of response is generally encouraged by unpredictable reinforcement that is gradually reduced. From this, it can be implied that there may be benefit in the system providing different types of reinforcement, depending on the child's stage of learning. This extends to having a continuous reinforcement strategy for younger and lower ability children, whilst providing an unpredictable schedule with older and higher ability children. Then between these ranges, an intermittent approach could be taken.

Learning is also linked to motivation because in order to persist with learning, an individual needs to be motivated. Motivation can be considered as 'an internal state that arouses, directs, and maintains behaviour' (Woolfolk 2004, p.350). Motivation comes in different forms; some individuals are motivated by intrinsic factors such as personal capabilities. For others, their motivation comes from extrinsic factors such as incentives, rewards and punishments. According to the Expectancy X Value theories (Tollefson 2000, Eccles and Wigfield 2001, cited by Woolfolk 2004), people will only be motivated if they believe they are capable of achieving a goal; there is value in achieving the goal and the cost of doing so can be justified.

Therefore, a software tutor should attempt to intrinsically motivate by allowing for personal improvement, however, this is difficult to quantify. Yet it is possible to attempt to extrinsically motivate through reward that is seen to be valued by the user group. It should also be remembered that the tasks required to achieve this reward should be within a suitable range such that the children believe they are capable of successfully completing them.

As previously discussed, another factor influential to learning is social interaction. Vygotsky believed that interaction with peers is essential for cognitive development. In view of this and because pair work at computers is a common in primary schools, specific consideration will be made to peer learning and collaboration between children.

2.3.4 Peer Learning

‘Peer learning is an educational practice in which students interact with other students to attain educational goals’ (O’Donnell and King 1999, p.3). It is seen as a way to prepare children for later in life when they are likely to be involved in co-operative work. When children work together, they can learn from each other and discuss concepts in familiar terms. According to De Lisi and Goldbeck, ‘it is likely, given the right setting, that children working in small groups or dyads will generate more sophisticated and creative answers to challenging questions than children working alone’ (O’Donnell and King 1999, p.35).

From a Piagetian perspective; when children work in collaboration with peers they are likely to come across conflict leading to disequilibrium. This provides the ‘necessary contexts for students to revise their current cognitive systems’ (O’Donnell and King 1999, p.5), in order to move back into equilibrium. Vygotsky’s zone of proximal development theory can also be applied to peer learning. Research suggests that pairing with a more competent partner can lead to improvement in the less competent partner. However, pairing with a less competent partner may lead to regression (O’Donnell and King 1999).

Peer work has to be carried out the right way because according to De Lisi and Goldbeck, ‘Having students work together versus alone is not enough to ensure that change in cognitive systems and performance occurs. Instead, the nature or quality of the peer interactions is crucial’ (O’Donnell and King 1999, p.37). In order to benefit from collaborative work, all group members need to participate.

According to Cohen (O’Donnell and King 1999), use of complex tasks that cannot be completed individually and require the combined expertise of the group are likely to maximise participation. Peer learning can obviously be beneficial in some situations and software to facilitate this type of learning could be useful if implemented to encourage active participation by all group members. A consideration of groupware and collaborative technology should be made if software is to be used to aid peer learning effectively.

2.4 Groupware and Collaborative Technology

Collaboration generally consists of communication and co-ordination; both of which are normal and important aspects of everyday life. Collaborative technology, often known as groupware, attempts to support these aspects of collaboration. However, the success of such systems is often linked to the type of collaborative environment. There can be difficulties and challenges when collaboration cannot take place in a face-to-face setting.

Groupware can be classified according to the temporal and physical location of the participants. Collaboration can either occur synchronously (at the same time) or asynchronously (at different times). Geographically, participants can be co-located (in the same place) or remote (in different places). For a software based tutor, it is appropriate that the children are in the same place at the same time.

Where collaboration is synchronous and co-located, face-to-face communication can occur, thus making collaboration easier. However, where collaboration involves computers there can be problems. This is because many computer applications only support input by one user and one device at a time. Stewart et al. (1998) observe that when collaborative software is used by partners with a single input device (i.e. a mouse), there are typically large differences in behaviour between the active mouse controlling partner and the passive non-controlling partner. These behaviours include fighting for control of the input device; lack of attention and frustration by the passive partner as well as less collaborative communication. However, with additional input devices, Stewart et al. (1998) observed enhanced collaboration.

Monteith (1998) notes that computers in primary schools are mainly used by more than one child at a time, however, this sort of use does not necessarily lead to effective collaborative learning and communication. The SLANT project (Monteith 1998) looked at how primary school children talked and worked together at the computer. It was observed that often, one child would appoint themselves leader; children with computers at home would become impatient by less computer literate children and there were heated discussions to do with making the task of operating the computer 'fair'.

Clearly there is a need to allow each collaborator to have their own input device if this sort of behaviour is to be avoided. KidPad (Benford et al. 2000) is a system that supports use of several mice to encourage collaborative storytelling with young children. From this, it was found that there needs to be an obvious and noticeably different effect from the collaborative actions of two children than that of individual's actions. In this way, the children can discover the positive benefits of collaboration and are more likely to work together.

Developing such systems can be difficult because usually there is a single input channel consisting of inputs from one mouse and one keyboard. When additional input devices are added and used at the same time, the effect is a converged input which is meaningless. Therefore, there needs to be some means of floor control, whereby a mechanism is in place to allow users to take turns. Alternatively, the Single Display Groupware (SDG) software model, as described by Stewart et al. (1999) could be used. SDG enables users to work simultaneously at a single computer display using multiple input devices with each device allocated a separate input stream.

However, developing SDG systems can cause further issues. The interface elements that communicate with the computer must handle multiple simultaneous users. It is necessary to distinguish which user the input came from and have uniquely identifiable cursors. Previous attempts at building SDG have proven difficult, with significant time needed to develop the underlying mechanisms. Tse and Greenberg (2004), detail the SDGToolkit which enables rapid development of SDG applications and interface components. The SDGToolkit works by generating unique identifiers for each input device and linking input events to the device it was generated by. In the 2004 paper, Tse and Greenberg state their intent to extend the toolkit to work with other input technologies, including display surfaces that recognize multiple touches.

A further problem with this type of system is that users may have difficulty seeing what they are doing when the interface is shared by others. To solve this problem, a large interface, such as an interactive whiteboard could be used. These are commonly found in classrooms so it is appropriate that they are investigated further.

2.4.1 Interactive Whiteboards

Interactive whiteboards are technology suited to synchronous co-located group collaboration. They are large touch sensitive boards that are connected to a computer and digital projector (Becta 2003). The image from the computer is projected onto the whiteboard and the computer can be controlled through interaction with the whiteboard using a special pen or by touch, depending on the type of whiteboard. According to Becta (2004), interactive whiteboards are rapidly becoming a common classroom resource; they can be a powerful teaching tool when used effectively and can encompass and extend a range of teaching styles.

With the government setting aside substantial funds for multimedia resources in schools between 2002 and 2006 (Curriculum Online 2005), the use of interactive whiteboards is likely to increase. It is important to have software that can be effectively used on interactive whiteboards to help children learn. Software and resources for interactive whiteboards are currently available for free and to buy for key stage two mathematics and other subjects. However, they are mostly resources for teachers, allowing some interaction by the children, rather than software intended for autonomous use by an individual child or group of children.

As interactive whiteboards are being used increasingly within schools, it is necessary to find out how their use has been perceived by school children so far. Research of this type has been carried out with the Interactive Whiteboard Evaluation, sponsored by the Centre for British Teachers (Hall and Higgins 2005). In this evaluation, 72 students were interviewed in focus groups across the UK. From this evaluation, it can be seen that interactive whiteboards are viewed very favourably by teachers and students. Students think that interactive whiteboards are more versatile than plain whiteboards; they enjoy the multi-media capabilities and they believe interactive whiteboards make lessons more enjoyable and fun.

However, the research also identified problems. These include technical issues such as 'freezing and 'crashing' which can result in disruption, delay and frustration in lessons. Some students commented on the need to calibrate the whiteboards as problematic. Sunlight shining on the boards impairing visibility was another common problem. Also, most student groups felt that they should be allowed to use the interactive whiteboards more. It is suggested that access is teacher directed, with limited opportunities to interact during lessons.

Further to these problems, there are several other constraints when using interactive whiteboards. Rekimoto (1998) comments that text entry can be difficult as writing with an electronic pointing device results in text that is difficult to read and unrecognisable to the computer. Furthermore, existing whiteboards are designed to be used by one user at a time which prohibits parallel activities among collaborators.

If collaborators were able to use more than one input device at a time, there may be physical limitations related to sharing the space in front of the whiteboard. Also, shadows are cast on the screen by the person actively participating with the whiteboard, so if the number of active participants increased, this problem would be compounded. Furthermore, the view of the interactive whiteboard would be restricted for collaborators not directly interacting.

As a response to these problems, Rekimoto (1998) proposes the use of palm-top computers such as a Personal Digital Assistant (PDA) by each participant as a means of distributing the whiteboard functions. In this way, the user can control the whiteboard from the PDA or prepare new information for the whiteboard. The PDAs and whiteboard are connected by a wireless network and data can be easily and directly transferred between them using a facility known as 'pick-and-drop', which has similarities to drag-and-drop.

It is clearly possible to use other computer peripherals in conjunction with an interactive whiteboard. A wireless keyboard and mouse could be used alongside or to replace the use of the specialist pens provided with the boards. If each child was provided with a wireless mouse and keyboard and the software was set up according to the Single Display Groupware model, this may provide an effective means for all children to interact with the whiteboard simultaneously. Furthermore, voting devices could be used to answer questions on an interactive whiteboard but this may foster competition rather than promote collaboration.

These responses obviously rely on the availability of appropriate hardware and technology, which may not be feasible in schools due to funding limitations. Therefore, typical methods of floor control involving physical hand-off of the whiteboard pen between children may have to be used. However, there are other areas of design which can and should be focused on so that software for use on interactive whiteboards is usable. According to Rekimoto (1998), the large display surface of a whiteboard makes traditional GUI design ineffective, so different approaches are needed.

2.5 Usability Guidelines

Usability guidelines can be used to 'guide' design. According to Preece et al. (1994) there are two types of guidelines: high level *principles* and low-level detailed *rules*. Shneiderman (1998) details eight principles that he believes are applicable in most interactive systems. These principles can be found in Appendix A and should be considered when designing the tutoring system. Nielsen (1994a, cited by Preece et al. 2002) also presents a set of ten usability principles known as *heuristics* because they are applied principles. These heuristics are commonly used in heuristic evaluation, where experts evaluate whether a user interface conforms to them. These principles can be found in Appendix B.

These principles are not necessarily applicable to all domains. As such, domain specific guidelines should be considered. In order to design a usable educational system for children to be used on interactive whiteboards, usability guidelines for large interfaces; groupware and collaboration; children and educational activities should be investigated. However, with such disparate areas, it is likely that there will be conflicts in the advice given, therefore, only the most appropriate and complementary guidelines should be followed.

2.5.1 Guidelines for Large Interface

There is limited literature that covers usability heuristics for large interfaces and specifically interactive whiteboards. Somervell et al. (2003) report on the current developments of usability heuristics for large screen information exhibits. Their work can be applied to the scope of this project, as systems of this type are used to present information to users on large screens in a way that can be understood by the users. However, with these types of systems, the users have no opportunities for interaction.

The heuristics, as detailed by Somervell et al. (2003) that can be applied are:

- **Use appropriate colour schemes for supporting information and understanding** - Use cool colours for background or borders and warm colours for emphasis
- **Layout should reflect the information according to its intended use** - Delegate screen space according to information importance
- **Judicious use of animation is necessary for effective design** - Multiple, separate animations should be avoided
- **Use text banners only when necessary** - Reading text on a large screen takes time and effort. Try to keep it at the top or bottom of the screen. Use large sans serif fonts
- **Avoid the use of audio** - Audio is distracting and can be detrimental to others in the setting

2.5.2 Guidelines for Groupware

Gutwin and Greenberg (2000) propose that there are a common set of small-scale actions and interactions that occur during collaboration. They refer to these as the mechanics of collaboration and claim that groupware usability problems are usually caused by insufficient and mismatched support for these mechanics. The full set of mechanics can be found in Appendix C. Based on these mechanics and Nielsen's heuristic evaluation methodology, Baker et al. (2002) have introduced a set of groupware heuristics. Of these, the following are considered pertinent:

- **Provide Protection** - Concurrent activity is common in shared workspaces; people should be protected from inadvertently interfering with the work of others
- **Manage the transitions between tightly and loosely coupled collaboration** - Coupling is the degree to which people work together; the movement between individual and group work should be fluid
- **Support people with the coordination of their actions** - Facilitate turn taking and negotiating the sharing of the common workspace

Clearly these heuristics apply to a range of groupware systems, however, where the groupware is to be used by children in a learning context, further principles can be applied. A key idea is that such software should encourage *exploratory talk*, where partners engage critically but constructively with each other's ideas (Mercer 1995, cited by Monteith 1998). Findings from the SLANT project (Monteith 1998), have resulted in design principles to encourage exploratory talk, these include:

- **Make sure problems are sufficiently complex** – In order to benefit from multiple perspectives
- **Provide props for reasoning** – E.g. supporting information
- **Don't let them turn it into a competitive game** - Time measures and small discrete tasks should be avoided as they may encourage competitive turn taking
- **Discourage mental turn taking** - Children should discuss what they are doing rather than act separately with responsibility for their choices
- **Encourage role-play and narrative** - This tends to result in lengthy discussion
- **Minimise typed input** - This is seen as a daunting task for most children which can take over the direction of the whole exercise

2.5.3 Guidelines for Educational Software

Lewis et al. (1998) detail adapting user interface design methods for the design of educational activities. Relevant principles from this research include:

- **Incorporate guiding knowledge into the task specification, not just into the background information** - Knowing what the specific task is allows very specific guiding knowledge to be attached to it
- **Design in failure** - It may be necessary to minimise the likelihood the user will find an efficient way to perform the task, thus forcing the learner to solve the problem in a way they would not encounter if the task were too easy
- **Design to avoid generating misconception** - Avoid presenting misleading models

2.6 Understanding Users and Interaction Styles

Although the guidelines discussed may provide valuable advice for designing a usable peer learning mathematics tutor, they are not sufficient in isolation. It is necessary to understand users so that only the relevant advice is incorporated into the design. Furthermore, the style of interaction used should be chosen based on the needs and capabilities of the users.

2.6.1 User Experience

One of the key considerations about users is that of their experience. Shneiderman (1998) classifies users into three different groups according to their experience, these are:

- **Novice or first-time users** - Novice users are assumed to know little of the task or interface, whereas first time users know the task but little of the interface. Issues with this group relate to anxiety with using the system. Therefore, the system should attempt to reduce this by using consistent, familiar and limited vocabulary, coupled with small confidence building actions in addition to informative feedback
- **Knowledgeable intermittent users** - Users in this group understand the task and interface, but have difficulty remembering specific features. It is necessary for the system to reduce their cognitive load by providing order, consistency and a focus on recognition rather than recall
- **Expert frequent users** - Users in this group are thoroughly familiar with the task and interface. For this group, the system should allow the user to get their work done quickly, often through keyboard shortcuts

Children in key stage two are most likely to fit into the first two categories. Clearly, on first use of a software based mathematics tutor they will be new to the tasks and interface. However, through use of the system, their experience will increase and it is likely that they will become knowledgeable intermittent users. However, they are unlikely to use the system often enough to become expert users, due to time constraints and availability of computers.

Shneiderman (1998) points out that it is easy to design for one class, but difficult to design for several. He suggests using a level-structure, so that novices are given a 'training-wheels' interface, before gradually progressing to more complex tasks. With a software based mathematical tutoring system, tasks could be considered as those involved with interacting with the system and those related to mathematical learning. Clearly the latter will need to increase in difficulty as the user's learning progresses. This could be done through a level

structure that corresponds to the different years or ability levels in key stage two. However, in terms of interaction tasks, it may not be appropriate to increase the complexity. Given that the users are children, it is probably better to keep the interface simple, consistent and familiar. Also, the system should attempt to reduce the user's cognitive load through recognition rather than recall so that the user can concentrate primarily on the learning task.

2.6.2 Interaction Styles

In view of the users and their experience, it is necessary to make decisions on the type of interaction the system should use. Shneiderman (1998) details five types of interaction styles; these include menu selection, form fillin and direct manipulation. Aspects of menu design may be appropriate, for example, by giving the user a choice of navigation buttons or for questions with multiple choice answers. Furthermore, form fillin may be suitable if used in the right context, for example, to enter an answer in a learning task. However, it should be remembered that form fillin is not really appropriate for novice users, therefore, it should be used judiciously. Over and above these, direct manipulation is the most obvious and appropriate style of interaction given the users' level of experience and stage of cognitive development. Furthermore, the visual representations of actions and objects allow abstract concepts to become concrete, which as already discussed, is particularly significant where children are in Piaget's concrete operational stage of cognitive development.

Shneiderman believes the 'trick in creating a direct manipulation system is to come up with an appropriate representation or model of reality' (1998, p.201). In doing this it is common to make use of metaphors. One of the best known metaphors used in software is that of the interface metaphor based on the physical office, as pioneered by Xerox in the 1980s (Smith et al. 1982, cited by Preece et al. 1994). However, this metaphor may not be appropriate for children as an office is not necessarily familiar to them. Instead, metaphors should be based on concepts children understand and are familiar with. This leads to a clear need to involve users in order to find out what they know, are familiar with and are interested in.

2.7 Involving Users

The main reason for involving users is so that designers can develop usable systems. According to Preece et al. (2002), a usable system is easy to learn, effective to use, and provides an enjoyable user experience (Preece et al. 2002). Furthermore, a usable system can be considered as one that 'supports, extends and positively transforms the work of the individuals, teams and businesses' (Holtzblatt and Jones cited by Schuler and Namioka 1993, p.177). To achieve this, firstly the right users need to be identified, and then their needs must be transferred into a usable system.

However, according to Schuler and Namioka (1993), too often systems are unusable; they do not work; or fail to do things expected of them. Furthermore, 'many computer systems adversely affect the performance of the workers using them' (Schuler and Namioka 1993, p.30). Where systems do not meet the needs of the users; the system may be unused and rejected or it may result in costly post-implementation maintenance improvements. Unusable systems usually arise when the needs of the users have not been fully considered. This may be because the users have been incorrectly identified or not involved at the right level.

To involve users, designers are faced with many challenges. Although they may realize the importance of user involvement; it is not always practical or possible to involve users extensively due to the resource constraints of the design team or the users. Where user involvement is limited, the designers must make assumptions about the users and how they will use the system in order to fill in the gaps. These assumptions will be based on the designers' implicit knowledge and judgments about the users and their work environment. Where users are involved, it is not always effective due to the methods used or because of misinterpretations made by the designers around the information collected.

There are many methods that can be used individually or in combination to involve users. The methods used should be chosen based on typical user characteristics, access to users and limiting factors. For this project, the users are key stage two children and teachers, however, teachers are likely to be indirect users. Access to them is likely to be through a primary school, therefore, time and access may be limited. Also, there may be legal and ethical concerns around the extent and nature of involvement. Furthermore, children may not understand what they want a system to do because they cannot envision how it will be used. If they do know what they want, they may be unable to articulate this. Therefore, it is necessary to choose appropriate methods and adapt these suitably.

2.7.1 Ethnography

Ethnography is a useful approach to understanding users. According to Blomberg et al. (Schuler and Namioka 1993, pp.123-127), 'Ethnography involves understanding the world from the point of view of those studied', through 'a period of field work where the ethnographer becomes involved in the everyday activities of the people studied'. There are four main principles that guide ethnographic work, these are:

- **Natural Settings** - Field work should be carried out in the everyday natural settings of the people being studied
- **Holism** - Context is important and behaviours can only be understood in the everyday context in which they occur
- **Descriptive** - Understandings about the group studied and the way the people actually behave are descriptive and non-judgemental
- **Members' Point-of-view** - The point of view of those studied must be understood, behaviours should be described in relevant and meaningful terms

Ethnography can provide useful information because often what people say and do are not the same. People may be unable to explain what they do as they 'often don't have access to the inarticulated, tacit knowledge associated with certain activities' (Schuler and Namioka 1993, p.130). Through observation, discussion and interviews, an ethnographer can build a clear contextual picture of the group being studied.

Observation can be carried out in an unobtrusive way, where the field worker does not interact with the people being observed. In this way, thorough notes can be taken, however the field worker may not fully understand the situation being observed. Conversely, the field worker can become a full participant in the activities, giving them first hand experience at the expense of concurrent note taking. Observations are often coupled with interviews and informal discussions. In the early periods of field work, interviews should be unstructured and open-ended, to allow participants to shape the discussion because 'asking highly

constrained questions before enough is understood about the situation likely will produce poor quality answers' (Schuler and Namioka 1993, p.134).

The information gathered through a period of ethnography can be used to aid the design of a system for the group being studied. The best results are likely to come from studies that involve designers, because the study will be more focused on the design task and the designers will have a better understanding of the users. However, ethnography is not always appropriate. It is a time consuming process and it is difficult to know when you have observed enough. Bloomberg et al. (Schuler and Namioka 1993, p.132) believe that when the behaviour can be predicted, based on repeated observations, it is likely that enough observations have been made.

If ethnography is to be used to understand children in key stage two, significant periods of observation would be needed within a classroom context, coupled with interviews and discussions with children and teachers. A full ethnographic study is not necessarily practical or appropriate due to time constraints and access to a school. However, aspects of ethnography could be applied in a limited way and combined with other methods.

2.7.2 Participatory Design

With participatory design, users are actively involved in the design process; they are an equal part of the design team which allows them to have an increased sense of ownership for the product being designed (Preece et al. 2002). The approach is iterative: the design is subject to evaluation and revision at each stage (Dix et al. 2004). Participatory design can involve several methods each of which have a common theme of providing a 'common language' for users and designers to communicate. This is essential if users' needs are to be understood by designers. Typical methods used in participatory design include brainstorming, storyboarding, PICTIVE and CARD.

Brainstorming

Brainstorming can be used to gather ideas from participants. It is usually carried out informally and without structure. However, Santanen et al. (1999) discuss the use of a technique called directed brainstorming to enhance creativity in group brainstorming sessions. Directed brainstorming involves giving prompts to direct participants to focus their efforts and thoughts in specific ways. It is suggested that stimuli from different contexts may lead to the exploration of new or different areas of knowledge networks.

CARD

CARD (Collaborative Analysis of Requirements and Design) uses prepared cards with system components on them to explore workflows and understand the macroscopic view of a system (Muller et al. 1995, cited by Preece et al. 2002). CARD supports equal collaboration and communication amongst stakeholders in a 'common language'. The simple low-tech physical form of the cards allow all participants equal access to them, and equal facility in moving, re-arranging, modifying and making new cards (Lafrenière et al. 1999).

As described by Muller (2001), CARD sessions are generally informal with an open and non-judgemental atmosphere. The session usually starts with an introduction of all participants, followed by an introduction of the materials. The participants use the cards to lay out a sequence of activities describing what is being done and why. The components on the cards

can be a work activity, an object in the workplace, a person, an interpersonal activity, or a mental operation or state. Muller (2001) advocates using different coloured cards to indicate a high-level class hierarchy, with related cards using the same colour. The outputs of the session can be the ordered cards as well as improved team relationships and commitment.

PICTIVE

PICTIVE (Plastic Interface for Collaborative Technology Initiatives through Video Exploration) uses basic stationary to design interactively on a shared design surface (Muller 1995, cited by Preece et al. 2002), whilst being recorded by video to capture the detail and ideas from the session. The resulting artefact is a low-level pen and paper prototype that represents the collaborative ideas of all participants.

The PICTIVE method is best applied when all participants can prepare before the session. The users are asked to prepare concrete and specific ideas about what they would like the system to do for them. The developers are also asked to construct a preliminary set of system components, for example, menu icons.

PICTIVE is a useful method because it is an ‘equal opportunity’ method with a shared language ‘that facilitates communications across different domains of competence’ (Schuler and Namioka, 1993, p.224). In this way, users and developers can get together to express their ideas directly in terms that are familiar to all of them. This is particularly pertinent where the users are children, because there will be different domains of competence.

It is likely that PICTIVE is suitable for use with children because, according to Muller (Schuler and Namioka 1993, p.227), ‘PICTIVE is deliberately informal and toy-like, which results in an atmosphere of play, in which people draw pictures, cut out shapes, and design new components on an experimental, exploratory basis’. Druin (1999) also believes that participatory design can be used with children; however she thinks the methods need to be adapted and changed to suit the children. On the other hand, some people question whether children are capable of contributing throughout the research and development process (Scaife et al. 1999, cited by Druin 1999).

One area of contention with PICTIVE when children are involved is the video recording element as there may be issues of consent. Furthermore, Druin (1999) notes that children tend to “perform” or “freeze” when being recorded. Therefore, the PICTIVE method may have to be adjusted by carrying out audio recording instead. With audio recordings, consent is more likely to be given; the children may feel more comfortable and the transcription time will be reduced. This is clearly better than note taking alone, which according to Druin (1999) can make children feel uncomfortable and distracted.

2.7.3 Contextual Inquiry

Contextual inquiry, as described by Holtzblatt and Jones (Schuler and Namioka 1993), is a technique that fosters participatory design. It follows three principles:

- **Context** - Talking to people in their actual work environment, where they are around the artefacts of work. The idea is to get people to articulate their work experience as they work, so they can be more concrete, rather than talking in abstractions as would happen

in traditional interviews. By using paper prototyping in actual work contexts, users can imagine the effects of a potential system on their work

- **Partnership** - To have an effective dialogue, partnerships must be formed with the users. It must be recognized that the user is expert in their domain; therefore, it becomes acceptable for designers to ask questions
- **Focus** - Having clarity of the domain of work being explored allows a more purposeful and effective conversation with the user. This focus can then be expanded where appropriate to cover areas more information is needed about

When carrying out a contextual inquiry, firstly the ‘customers’ should be identified, then a framework for the visit should be developed. A visit usually takes place over several hours, with several interviewers. The visit normally starts with an introduction session to establish relationships and set the scene for the visit. This is followed by ongoing work inquiry where the user works whilst being asked open-ended questions; thus allowing the user to lead the discussion. The visit is concluded with a wrap-up session to clarify uncertainties and make summaries about what has been learned. The information gathered from a contextual inquiry can then be used to inform the design.

The relation to concrete artefacts in the context principle and the use of paper prototyping is clearly applicable for use with children, particularly those in Piaget’s concrete operational stage of development. Also, having a clear focus should allow good use of the time available with the users.

2.7.4 Co-operative Inquiry

Building on participatory design and contextual inquiry, Druin (1999) describes co-operative inquiry as an approach to support intergenerational design teams in developing new technologies for children, with children. Co-operative inquiry is based on the ideas that multidisciplinary partnerships with children are needed; field research should be carried out to understanding context, activities and artefacts and iterative low-tech and high-tech prototyping should be used. Co-operative inquiry is composed of the following techniques:

- **Contextual inquiry** - Adults and children are researchers with their own styles
- **Participatory design** - Children and adults work together to produce low-tech prototypes, allowing children to have a voice
- **Technology immersion** - Children are observed to see how they use technology over concentrated periods of time

Interesting findings from this research include the observation by Druin that adults can have difficulty interacting with children due to traditional ‘power structures’ with adults acting like authority figures. Adults may not know how much control to release to the children. Druin believes children and adults must work and make decisions together. She advocates wearing informal clothing to appear more like a peer.

Furthermore, Druin has found that children aged 7-10 years old make the most effective prototyping partners as they are verbal and self reflective enough to discuss what they are thinking; can understand abstract ideas of low-tech prototyping and aren’t too heavily burdened with pre-conceived notions of the way “things are supposed to be”. In view of this it is expected that children in key stage two will be able to make valuable contributions.

2.7.5 A Mixed Approach

Several methods of user involvement have been discussed, with some mention as to how these can best be transferred to situations where the users are children. In view of this research it has been decided that a mixture of methods will be used in this project. In terms of ethnography, a limited approach will be taken, ideally with several lesson observations, teacher interviews and children interviews. This approach clearly resembles that of a contextual inquiry as it will occur in the context of a school, with an emphasis on building relationships and with a clear focus in order to make the most of the time available. However, the process may have to occur over several disjoint sessions, whereas contextual inquiry usually takes place in one extended session.

This initial involvement should provide an understanding of the users, so that the next stage in user involvement will be around design of the mathematics tutoring system. For this, several methods from participatory design will be used, starting with directed brainstorming, followed by CARD then PICTIVE. In such a way, design will move from initial ideas, to a high-level understanding of the system before going into detailed low-level aspects of the system. Additionally, the findings from co-operative inquiry relating to involvement with children in intergenerational design teams will be applied in order to suitably adapt the participatory design methods.

2.8 Conclusion

Through this literature survey it has been possible to get a better understanding of the problem domain and the associated issues. At a high level, the problem can be considered to consist of two parts. The first relates to developing a system that adds value to a child's learning and development in a peer learning context. The second relates to making this system usable and enjoyable for the identified user group.

In terms of this learning aspect, it is possible to use the research on cognitive development, learning, peer learning and motivation as the basis to designing a valuable mathematical tutoring system. A key finding from the research is that children in key stage two are most likely to be in Piaget's concrete operational stage of cognitive development, thus teaching should be linked to concrete objects and representations as the children may not be able to understand abstract problems. Furthermore, teaching should be related to concepts a child is familiar with so that they can attempt to assimilate new information into existing memory schemes. This could be achieved through use of suitable metaphors in the system. Furthermore, Vygotsky's zone of proximal development is highly relevant as it may be possible to enhance learning if the system can provide suitable scaffolding within a child's problem solving range. To achieve this, use of an assistive interface persona is suggested.

The literature review has also highlighted a need for tuition to be tailored to a child's individual needs; however, it is known that computers in primary schools are rarely used individually. Thus the need moves to providing a system that is suitable and supportive of peer learning and collaboration. A system designed for peer or group use may actually enhance learning. Research has shown that peer learning can be beneficial and this is supported by Vygotsky's socio-cultural theories.

However, using technology can sometimes detract from collaboration and peer learning because most applications are designed to be used with one input device, which can cause user control issues. The single display groupware (SDG) model overcomes this by allowing multiple input devices.

Interactive whiteboards are particularly suited for use by groups as they have a large shared interface. There is potential to design a system for use on interactive whiteboards as they are seen favourably by pupils and teachers alike. Also, their use is likely to increase within schools. However, interactive whiteboards are usually controlled by one person at a time using whiteboard pens directly on the board, or through peripheral devices, making simultaneous user interaction impossible. Nevertheless, it may be possible to use voting devices or multiple wireless input devices using the SDG model. But there are limitations with this as this sort of technology is unlikely to be available within schools. Instead, tasks and content likely to foster collaboration and peer learning should be provided by the system.

Clearly there is a strong need to make such a system usable, so that the system does not get in the way of the user's learning. Usability guidelines can be used to influence design but care should be taken because there may be conflicting advice as the system spans different domains. Also, usability guidelines cannot necessarily advise about the user's interests and what motivates them. This is an important consideration because the system should be enjoyable to use and motivating enough to encourage persistence of use.

A better way to build in usability and find out what users like is through direct user involvement. Different methods for involving users have been discussed with relation to use with children, including participatory design. However, it has been shown that there is a need to adapt and combine methods in order to work around the limitations that are inherent to working with children. Furthermore, it is also necessary to involve teachers because there is some doubt as to what children can actually contribute and whether they are aware of what is best for them in a learning context.

This user involvement will begin in the requirements analysis phase of the project which will be detailed in the next chapter. This requirements chapter will also draw on the findings from this literature survey as well as evaluating existing relevant software systems with a view to generating requirements.

3 Requirements Analysis and Specification

3.1 Introduction

This chapter will detail the processes used to capture and analyse requirements. A requirement can be thought of as ‘a statement about an intended product that specifies what it should do or how it should perform’ (Preece et al. 2002, p.204). The requirements will be developed iteratively with a user-centred approach, as advocated by Preece et al. (2002). This involves ‘identifying needs’ then developing a stable set of requirements that form a sound basis to move towards design.

The requirements will be derived from a range of sources including the literature review, stakeholder involvement and evaluation of relevant existing software. The literature review has provided a basis for analysing requirements through the exploration of a range of theoretical and academic areas. However, this research is not sufficient in isolation; it is crucial to involve stakeholders if a system is to meet the users’ needs. Additionally, there is value in evaluating existing mathematical tutoring software so that positive aspects can be carried forward and poorly designed features can be specifically excluded. This will then lead to a requirements specification, which can be found in Section 3.3.

3.2 Requirements Analysis

To establish system requirements, a comprehensive understanding of the problem domain is needed and will be achieved by utilising a range of sources as discussed.

3.2.1 Literature Review

The literature review has covered a range of topics relevant to the problem domain, including the key stage two mathematics national curriculum, children’s cognitive development and peer learning. This research will influence requirements relevant to the content and functionality of the system, with a view to facilitating mathematical learning in a peer context. Also, the discussion on knowledge, reinforcement and motivation provides valuable information on how learning can be made more successful.

Furthermore, the discussion around collaboration and groupware, including interactive whiteboards, has identified potential issues that need to be considered if the system is to be used on this type of technology. Additionally, the experience of the users and usability guidelines relevant to the problem domain should be used to influence the requirements with a view to specifying a usable system.

The requirements sourced from the literature review can be found clearly identified in the requirements specification (see Section 3.3).

3.2.2 Stakeholder Involvement

According to Dix et al. (2004), a stakeholder is anyone directly affected by the success or failure of the system. The group intended to directly use the system are key stage two children (aged seven to eleven), which makes them the primary stakeholders. The system is anticipated to be used within schools, therefore, head teachers and teachers of key stage two children can be considered as secondary stakeholders. Teachers are unlikely to use the

system directly, however, their pupils may use the system, and therefore it should be aligned with standard teaching practices.

Additionally, the system will need to consider the national curriculum, thus making the Qualifications and Curriculum Authority (QCA) tertiary stakeholders because they maintain and develop the national curriculum. Other stakeholders that require consideration are facilitating stakeholders; these stakeholders include the people involved in design, development and maintenance of the system.

A system should attempt to meet the needs of as many stakeholders as possible; but this is not always possible where there are differences of opinion or conflict. The primary stakeholders' needs usually take precedence, however, if they are children it could be argued that they do not know what is best for them in a learning context or are unable to articulate their needs, making input from other stakeholders more important.

Moorlands Junior School in Bath has agreed to participate by allowing access to their school for lesson observations, teacher interviews and participatory design with a group of children from year 5 (aged nine and ten). The time that can be spent in the school is at the discretion of the school and is to some degree limited. Due to these limits and to address issues of generality, two children from a different location and school have been interviewed and involved in a pilot brainstorming session to provide additional primary stakeholder input.

Lesson Observations and Teacher Interviews

Lesson observations and teacher interviews have been carried out to find out more about current teaching styles and methods, with a view to transferring successful practices into requirements. Particular interest was given to where and how ICT, interactive whiteboards and other resources are used within the school, particularly for mathematics. A further area of interest was group work as the system is intended to support learning in a peer context.

Mathematics lessons were observed for years 3 and 4 (aged seven to nine); these are the first two years in key stage two. Observations for each year in key stage two would have been ideal, but were not possible due to constraints set by the school. However, in order to gather information from a wider range of primary stakeholders of different ages, children from year 5 will be involved in participatory design. A year 4 ICT lesson was also observed, with similar aims as for the mathematics lesson observations. However, by observing children using computers, a better understanding about their experience with computers and associated technology could be obtained.

Teacher interviews were carried out after the lesson observations to allow for clarification of aspects observed in the lesson. The interviews were semi-structured to guide the conversation over specific areas of interest, but with scope to allow the teachers to steer the conversation. By carrying out the interviews in the teacher's classroom, it was hoped that they would talk about artefacts and systems used there.

Year 3 mathematics lesson (14-Nov-2005)

A typical year 3 mathematics lesson was observed. The lesson started with the class seated around the interactive whiteboard and the teacher giving an oral explanation of the topic. The teacher had prepared questions (on ACTIVPrimary2 flipchart software) to use on the interactive whiteboard. A child would often be chosen to write their answer on the

interactive whiteboard and explain how they got to the answer. Later in the lesson the children worked individually on a series of questions at their desks. Following the observation, the teacher was interviewed.

Year 3 mathematics lesson observation - key findings and areas of interest:

- The teacher set clear objectives for the topic being covered at the start of the lesson
- The teacher provided content and questions for different mathematical abilities
- The teacher set clear expectations of the number of questions to answer and time available
- The teacher was available to help and prompt while the children tried the questions
- The teacher gave opportunity for the children to rate how they found the work
- The interactive whiteboard was located about one metre above the ground, making the top half of it unreachable for shorter children. Clearly, software for children intended for use on interactive whiteboards should not require interaction with the top of the screen
- The interactive whiteboard is often obscured by the shadow of the interacting child
- Some children had difficulty reading from the interactive whiteboard at their desks. This implies that text used in software for interactive whiteboards should be a suitable size and font to be read from a reasonable distance away from the screen
- Direct sunlight on the interactive whiteboard made the content difficult to see. This implies that software for interactive whiteboards should choose colour combinations as to attempt to minimise this problem

Year 3 teacher interview - key findings and areas of interest:

- The teacher tries to make lessons as interactive as possible
- The teacher thinks building confidence is important, that children should be encouraged to continue even when they have got something wrong
- The teacher thinks it is important to provide feedback to the children and allow them to give their own feedback about how they found the work
- The teacher aims examples at the middle ability, but provides extension and easier work for different levels
- The teacher thinks division is difficult to teach, so should be taught with multiplication
- The teacher uses the interactive whiteboard in most lesson and usually prepares the content herself
- The teacher thinks the children in year 3 are not necessarily competent using the interactive whiteboard because they have not been taught, but some are better than others
- The year 3 ICT work is not linked to the mathematics curriculum
- The teacher sometimes uses free games (interactive-resources.co.uk) on the interactive whiteboard
- The teacher sometimes divides the class into two teams when playing games
- The teacher thinks number related topics do not work well for group work, but measuring and capacity work well for group work due to their practical nature

Year 4 mathematics lesson (18-Nov-2005)

The observed lesson focused on assessments, with short multiplication club quizzes that are carried out daily, followed by a standard mental oral test that is carried out less often. Following the lesson, the teacher was interviewed.

Year 4 mathematics lesson observation - key findings and areas of interest:

- The interactive whiteboard displayed the test instructions and time remaining
- Children were trusted to mark their own work on the multiplication club tests
- The teacher provided reward for good work, in form of praise and joining the next level in the multiplication club, which was visible to the whole class
- The teacher provided encouragement for children who needed to improve in the multiplication club tests
- The teacher read out questions rather than playing a CD for the mental oral test because he thought the CD is unclear and recorded voices can be difficult for children to understand

Year 4 teacher interview - key findings and areas of interest:

- The teacher tries to make lessons as practical as possible, with a hands-on approach making use of resources including blocks to help with calculations
- The teacher thinks it is important to know what level the child is at so that appropriate examples and questions can be used for their level
- The teacher thinks testing children with standard questions, as with the mental oral test, can help to establish their level of mathematical ability and identify areas for improvement
- The teacher thinks a child can be demoralised if they cannot answer a question. So constructive feedback is needed, avoiding negative words like 'but' and 'although'
- The teacher thinks praise and encouragement is needed to build confidence
- The teacher thinks it is important to relate work to what children know about and are interested in, for example, animals
- The teacher thinks children often find division, ratio and proportion difficult
- The children do not use textbooks and the teacher thinks this is common in many schools
- The teacher said he uses the interactive whiteboard in about 90% of maths lessons
- The school mostly uses ICT for literacy but they will soon start using it for maths
- Children frequently work in pairs and groups of varying sizes. The teacher likes to partner children from neighbouring ability levels
- The teacher thinks word problems and those requiring a process or a number of stages are good for group work

Year 4 ICT lesson (29-Nov-2005)

The ICT lesson observed covered the topics of time and number sentences. An activity in a Microsoft Word file was used that has a series of analogue and digital clocks. The aim was to move the digital clocks to match the analogue clocks. Following the observation the teacher was interviewed.

Year 4 ICT lesson observation - key findings and areas of interest:

- The teacher demonstrated navigating to the Microsoft Word file in the network hierarchy of folders, then demonstrated the activity on the interactive whiteboard
- Most children had to work in pairs as there are only 16 PCs in the ICT suite
- Some pairs had one child using the mouse and the other on the keyboard. In some pairs, 'fighting' for control of the input device was noticed. Thus indicating the need for multiple input devices or a fair way of sharing the control

- The processes of logging on and navigating to the activity took some children a long time, therefore it could be implied that these tasks are barriers to children in achieving the intended tasks
- Software with extra functionality that is not needed to complete the task, as in Microsoft Word, can confuse children or cause them to make mistakes
- It should be possible to correct mistakes, for example with the 'undo' button in Microsoft Word, which the teacher referred to as the 'oops button'
- The children found the on-screen help characters in Microsoft Word interesting, but they were a distraction from achieving the intended task for some pairs

Year 4 teacher interview - key findings and areas of interest:

- The teacher acknowledged that activities in Microsoft Word have limitations, but the school cannot necessarily afford specialist software
- The teacher explained that there are a range of abilities in terms of computer use in the class, which is generally linked to academic ability and computer use at home
- The teacher explained that the children always work in pairs at the computer, as there are not enough for them to have one each. The teacher does not think this is a problem unless they need to save their work because they can only save to one network account
- The teacher tries to get the children to swap control of the computer half way through the lesson, so that both children get a chance to control the computer

Interviews with Children (26-Nov-2005)

Two children were interviewed; the first was a nine year old girl in year 5 at a school in Kent. The second was a seven years old girl in year 2 at a school in Kent. She was interviewed because she is in the last year of key stage one, so it was thought that her perspective may provide information about what children are like as they enter key stage two. Both interviews were semi-structured with the aims of finding out the children's opinions about school subjects, specifically maths; their thoughts on interactive whiteboards; their computers experience and their views on group work.

Year 5 child interview - key findings and areas of interest:

- Her favourite subjects are Art and Design, she likes creative subjects
- Her least favourite subject is Maths because she finds it difficult, particularly division because she "can't do it" and "doesn't understand it yet". However, she does like the times tables because she thinks they can be learned
- She thinks maths is taught well because her teacher does not use hard words
- She said she often has to wait for the teacher's help and would like more help
- She likes interactive whiteboards but has not had a chance to use one yet
- She does not normally use the computer for maths, apart from the calculator on her home computer when she "can't be bothered to work it out" in her head
- She uses the internet at home, mostly for entertainment e.g. music videos and games

Year 2 child interview - key findings and areas of interest:

- Her favourite subject is 'computers' because she thinks they can be used for many things
- She finds maths 'boring' and finds subtraction particularly difficult
- She thinks calculators and more help from teachers would make maths better
- Her older sister has been teaching her times tables, so she finds some of these easy

- Some children in her class use bricks, number lines or arrow cards to help with maths
- She has not really used the computer for maths before
- She likes interactive whiteboards because they do not need chalk and can be written on using a pen. However, she thinks they are difficult to use because “you have to press quite hard” and she thinks small children get left out because they cannot reach
- She uses the internet at home, mostly for entertainment e.g. looking at websites about things she is interested in

This section has detailed pertinent findings from lesson observations coupled with teacher interviews as to give a better understanding of what is important when teaching children in key stage two but also to identify some of the challenges. Further to this, two children have been interviewed to get their perspective on aspects such as mathematics and computers. The requirements sourced from this stakeholder involvement can be found clearly identified in the requirements specification (see Section 3.3).

Although this information from stakeholder involvement is important, it is also possible to learn from existing software systems, therefore, existing software will be evaluated next.

3.2.3 Existing Software/Systems Evaluation

Ideally, existing systems should be evaluated with children, however, at Moorlands Junior School ICT is not often used with Mathematics. Therefore it was not appropriate to carry out evaluations with children from the school. However, ‘expert’ evaluations of mathematics software that has potential for use within the school can be carried out. According to Preece et al. (2002) experts can inspect the user interface and predict potential interaction problems. Typically discount methods of heuristic evaluation and walkthroughs are used. With heuristic evaluation (Nielsen 1994a cited by Preece et al. 2002), a user interface is evaluated against a set of usability principles, known as heuristics. Walkthroughs involve an evaluator ‘walking through’ a task using a system whilst noting any problematic usability features.

Education City (2006) and Primary Games (2006) are online interactive educational resources with games developed in Flash for key stage two. Both are registered Curriculum Online content providers and therefore, are considered relevant and appropriate systems as schools can purchase games from them using eLearning Credits¹. Games from both websites have been evaluated with Sylvie Girard, a student on MSc Human Communication & Computing. The evaluations were run as informal walkthroughs, with Sylvie talking aloud about any potential usability issues. Additionally, questions based on heuristics were used as prompts; however the evaluations were not carried out specifically as heuristic evaluations.

Two games on Education City were evaluated; the first was ‘Dancing Robot’, where the player must fill in the blanks to complete a sum and win points to make the robot ‘dance’. The second was an English game, called ‘Slam Dunk’ where the player has to shoot basketballs into hoops to spell the specified word. Although this was an English game, the evaluation is still relevant because the findings can be applied to the requirements for the system that will be developed.

On Primary Games, the ‘Treasure Island’ game was evaluated. In this game the player has to locate the treasure by making use of signposts which are revealed to indicate how far away

¹ Government funding for multimedia resources

the treasure is. These sign posts are revealed when the player searches at a position on the map, so the player should be able to get gradually closer to finding the treasure by searching.

See Appendices D and E for screen shots of Education City and Primary Games respectively.

Education City evaluation - key findings:

- The button to access the login screen is not very obvious. It is coloured green, which is too close to the background colour and it is located in the bottom right hand corner, where the user is unlikely to look first. This indicates a need to have important buttons in prominent locations with high figure-ground contrast against the background
- The 'Back' button is positioned inconsistently across different screens. This indicates a need to keep buttons in the same position across multiple pages
- Buttons do not have tool tips, so a player may not understand a button's purpose. This indicates a need to have tool tips to aid the players understanding of the interface
- Right clicking the mouse brings up Flash's Context Menu. Although this has been limited it could still cause some confusion or disruption to interaction. This indicates the need to try and remove or restrict the menu from being shown if Flash is used
- The instructions cannot be accessed without leaving the game, which may cause disruption to game play. The instructions should be accessible at any time
- The game countdown timer does not add much value and there is no warning when time is about to run out. It would be better to count up or to not have a timer at all
- It is not obvious if 'Quit' means leave the current game or leave Education City
- The user was shown their score and progress in all games evaluated

Education City Dancing Robot game evaluation - key findings:

- The input of the missing number must be made to a separate box, it would be better if this could be entered directly into the sum where the number was missing
- There are no input restrictions; however, as the game tests multiplication up to 10 x 10, the player could be given a reminder if a certain range is exceeded
- Red has been used to indicate an incorrect answer and green a correct answer. These colours may not be distinguishable for a colour blind user
- The 'prize' of an animation was not visible for long enough. The player should be able to control when they move onto the next screen from the animation
- When a game has finished the system asks 'Again?', which could be misinterpreted. It would be better to ask 'do you want to play again?'

Education City 'Slam Dunk' game evaluation - key findings:

- The game is slow moving with animations for each shot at the basket. With such a long time between shots the player may forget what word they are trying to complete. The player should be able to control the speed of this. Indicating that animations should not get in the way of game play
- It is not possible to pause and resume game play. The game should allow the player to specify when they are given the next word so the player has more control over the game
- It is not possible to save a game; the player could be in the middle of a game and have to leave. To play again they would have to start a new game
- There is a speaker icon which when pressed will repeat the word, however, the player may not know what this button does. Tool tips should be used to explain this

- There should be a limit to the number of times the player can see or hear the word, otherwise the game may be too easy
- The instructions make no mention of how to use the mouse for controlling aim, how to hear the word again or how scoring is worked out. These should all be included.
- The certificate cannot be previewed before it is printed.
- If the player tries to shoot at a basket where a letter has already been placed the game will not allow this and will inform the player why. This is good as informative feedback is important.
- The real world representation are good with the baskets and characters

Primary Games ‘Treasure Island’ game evaluation - key findings:

- The language used in the instructions may be at an inappropriate level for younger children. It should be simplified, or there should be audio version
- The player must leave the game to read the instructions and there are no tool tips
- The ‘Quit’ button is too dark for the background, a high figure-ground contrast is needed
- The blue background behind the game is too dark and may be distracting.
- It is not obvious when a game has finished as the search button is not deactivated. The player should be directly asked if they would like to play a new game
- The pointing finger icon for the ‘Begin’ button is good

From these evaluations, some positive design aspects have been identified; however, the evaluations have mostly highlighted usability issues. From these findings, it is possible to specify requirements that will attempt to avoid such usability problems. The requirements sourced from these evaluations can be found clearly identified in the next section.

3.3 Requirements Specification

3.3.1 Specifying and Structuring the Requirements

Requirements can be classified in different ways. As a user-centred approach to analysis has been taken, the requirements will be specified as ‘user requirements’. According to Sommerville (2001, p.98), these are high level abstract statements, written in natural language to define ‘what services the system is expected to provide and the constraints under which it must operate’. Preece et al. (2002) advocate using functional and non-functional classifications, with non-functional requirements divided into further categories. A subset of these categories will be used as some of these are more applicable to the system than others.

In view of this, the requirements will be specified using the following categories:

- **Functional requirements:** What the system should do
- **Non-functional requirements:** Constraints on the system and its development
 - **User Requirements:** Characteristics of the intended user group
 - **Usability Requirements:** Usability goals and measures

Each requirement will be structured in a specific way. Preece et al. (2002) recommend using the Volere Template Requirements Shell for guidance. This shell contains several fields and requires a significant amount of information for each requirement, however, for this project, a subset of these fields will be used as it is felt that some fields are not essential.

The fields that will be used are as follows (from Atlantic Systems Guild Limited 2006):

- **Requirement:** A unique identification number for the requirement
- **Description:** A one sentence statement of the intention of the requirement
- **Rationale:** A justification of the requirement
- **Source:** Where the requirement has come from

A clear distinction will be made between mandatory and desirable requirements. According to (Sommerville 2001), the common convention for defining mandatory requirements is with 'shall' and desirable requirements with 'should'. Therefore, this convention will be followed in the requirements specification.

3.3.2 Scope

It is necessary to define the scope of the requirements specification so that there are clear boundaries around what the system should and should not provide. Although peer learning has been a key consideration so far, for a system to explicitly support this, it is likely to need multi-user functionality with use of multiple simultaneous input devices. However, as mentioned, this may not be feasible given the intention for the system to be used in schools. Furthermore, to achieve this, the Single Display Groupware Model (see Section 2.4) may be necessary, thus greatly increasing the system's complexity beyond a range feasible to complete given the time available. In view of this, the system should instead support peer learning implicitly, by providing content likely to encourage collaboration between peers. This content could then later be extended to become an inherent part of a SDG system.

3.3.3 Functional requirements

Requirement:	1
Description:	The system shall provide the user with mathematical instruction linked to the key stage two mathematics national curriculum
Rationale:	The national curriculum defines the content and targets for mathematical teaching in schools. Vygotsky believed instruction is essential to reach the highest levels of thinking.
Source:	Literature Survey: 2.2, 2.3.2

Requirement:	2
Description:	The system should provide a suitably enjoyable learning experience
Rationale:	Enjoyment of learning is normally needed if learning is to be successful.
Source:	Literature Survey: 2.2

Requirement:	3
Description:	The system shall provide the user with clear objectives about the purpose of the tasks to complete
Rationale:	Teachers often set clear objectives in real mathematics lessons so that the children know the lesson purpose and what they should achieve, giving them a target. A principle for good learning and teaching is to structure and pace teaching so children know what is to be learnt, how and why.
Source:	Literature Survey: 2.2

	Stakeholders: Year 3 mathematics lesson observation
Requirement:	4
Description:	The system should provide a suitably challenging learning experience
Rationale:	Learning should be 'a challenging experience', so that disequilibrium can occur, causing the learner to adopt more sophisticated ways of thinking to reach equilibrium. However, the challenge should be within their zone of proximal development. Complex problems may benefit from multiple perspectives, making collaboration more likely.
Source:	Literature Survey: 2.2, 2.3.1, 2.3.2, 2.3.4, 2.4, 2.5.2, 2.5.3

Requirement:	5
Description:	The system shall provide the user with opportunity to practice
Rationale:	Repetition is often necessary to retain information permanently. Children need opportunities to practice.
Source:	Literature Survey: 2.3.3

Requirement:	6
Description:	The system shall clearly show the user's progress in task completion
Rationale:	So children can monitor their progress and know how well they are doing.
Source:	System Evaluations: Education City

Requirement:	7
Description:	The system should allow the user to save their progress in the system and resume use of the system at this saved point
Rationale:	The user may have to exit the system before having completed all tasks. They should be able to save and resume at another time.
Source:	Stakeholders: Year 4 ICT lesson observation System Evaluations: Education City

Requirement:	8
Description:	The system shall provide constructive feedback that guides the user away from making the same mistakes again, avoiding negative words
Rationale:	Teachers give constructive feedback so children know what they did wrong and how to improve by learning more appropriate strategies. Negative words like 'but' and 'although' should be avoided because they can be discouraging.
Source:	Stakeholders: Interview with year 3 teacher, interview with year 4 teacher Literature Survey: 2.3.3

Requirement:	9
Description:	The system shall allow the user to 'try again' for a limited number of times on incorrect completion of a task
Rationale:	The user may accidentally make an unintended action resulting in the incorrect completion of the task. It should be possible for them to try again to complete the task, but this should be limited otherwise the task may become too easy.
Source:	Stakeholders: Year 4 ICT lesson observation

	System Evaluations: Education City
Requirement:	10
Description:	The system shall provide visual learning information
Rationale:	It is believed that information coded both visually and verbally is easier to learn, however, audio output is not necessarily appropriate in a shared environment, such as a classroom, so only visual information should be used.
Source:	Literature Survey: 2.3.3, 2.5.1

Requirement:	11
Description:	The system shall provide relevant learning assistance to the user through an interface persona
Rationale:	If the user does not understand the task or how to complete the task, this may be a barrier to their learning progression. Assistance through scaffolding is needed to support the learner while they build a firm understanding until they can problem solve alone. In real lessons, teachers are usually around to help children if they get stuck. An interface persona could replicate adult or peer support.
Source:	Literature Survey: 2.3.2, 2.5.3 Stakeholders: Interview with year 3 teacher, Interview with year 5 child, Interview with year 2 child

Requirement:	12
Description:	The system should provide usability instructions that are accessible at any point in the system
Rationale:	Having to leave the system or tasks to read instructions may be disruptive to the learning task.
Source:	System Evaluations: Education City, Primary Games

Requirement:	13
Description:	The system shall provide reward to the user on correct completion of learning tasks
Rationale:	Praise and reward can be considered as reinforcement, which can aid learning. Teachers often give praise and reward for good work in real mathematics lessons. For some, motivation comes from rewards.
Source:	Literature Survey: 2.3.3 Stakeholders: Year 4 mathematics lesson observation

Requirement:	14
Description:	The system should give rewards valued by the user
Rationale:	According to the Expectancy X motivation theory, a key factor in motivation is the belief that there is value in achieving a goal. Rewards should be seen to be valued by children in key stage two.
Source:	Literature Survey: 2.3.3

Requirement:	15
Description:	The system shall not use timed learning tasks
Rationale:	Time measures can make tasks competitive, thus are not suited for peer learning and collaboration. Timed games do not always add value.
Source:	Literature Survey: 2.5.2 System Evaluations: Education City

3.3.4 Non-functional requirements

Requirement:	16
Description:	The system shall be able to be used on an interactive whiteboard
Rationale:	In order to facilitate peer learning and collaboration. Interactive whiteboards are often used in primary schools and their use is likely to increase. Interactive whiteboards can potentially make lessons more enjoyable and fun.
Source:	Literature Survey: 2.4, 2.4.1

Requirement:	17
Description:	The system shall be able to be used on a PC
Rationale:	In order to use the system on an interactive whiteboard, the system will also need to function on a PC.
Source:	Literature Survey: 2.4, 2.4.1

Requirement:	18
Description:	The system shall be able to function without a hardware keyboard
Rationale:	If the system is to be used on an Interactive Whiteboard it may not be practical to use a hardware keyboard.
Source:	Literature Survey: 2.4.1

Requirement:	19
Description:	The system shall function on hardware used in Moorlands Junior School
Rationale:	The system is initially intended to be used in Moorlands Junior School, therefore, it should be able to work on computer hardware used there.
Source:	Stakeholders: Year 4 ICT lesson observation

Requirement:	20
Description:	The system shall function on the operating system used in Moorlands Junior School
Rationale:	The system is initially intended to be used in Moorlands Junior School, therefore, it should be able to work on the operating system used there.
Source:	Stakeholders: Year 4 ICT lesson observation

3.3.5 User requirements

Requirement:	21
Description:	The system shall have content for different years in key stage two
Rationale:	Key stage two spans four school years, each year has different objectives, therefore the content should be tailored to these objectives.

Source:	Literature Survey: 2.2
Requirement:	22
Description:	The system shall have content appropriate for different mathematical ability levels within the different years in key stage two
Rationale:	Within each year group are children of different mathematical abilities. Content and questions need to be appropriate for the child's ability in view of the problem of the match. Tasks and content will need to increase in difficulty in line with increasing ability levels.
Source:	Stakeholders: Year 3 mathematics lesson observation, interview with year 3 teacher, interview with year 4 teacher Literature Survey: 2.2

Requirement:	23
Description:	The system should provide tuition linked to familiar concepts
Rationale:	By drawing on prior knowledge a learner can more readily develop an understanding of a new domain, as occurs with assimilation. Connecting new information to existing knowledge is often necessary to retain information permanently.
Source:	Literature Survey: 2.3.1, 2.3.3

Requirement:	24
Description:	The system should provide tuition linked to concrete concepts
Rationale:	Children in key stage two are likely to be in Piaget's concrete operational stage, meaning their problem solving is linked to physical reality and they cannot necessarily deal with abstract problems.
Source:	Literature Survey: 2.3.1

Requirement:	25
Description:	The system should use vocabulary familiar to the user
Rationale:	Meaningful teaching uses vocabulary that makes sense to the learner.
Source:	Literature Survey: 2.3.3 System Evaluations: Education City, Primary Games

3.3.6 Usability requirements

Requirement:	26
Description:	The system shall have a simple interface
Rationale:	As to minimise the user's cognitive load thus allowing the user to concentrate on the learning task.
Source:	Literature Survey: 2.6.1

Requirement:	27
Description:	The system shall primarily use direct manipulation
Rationale:	This is most appropriate given the users will be novice or first-time users going to knowledgeable intermittent users after having some experience with the system.
Source:	Literature Survey: 2.6.1, 2.6.2

Requirement:	28
Description:	The system should make use of menu selection where appropriate
Rationale:	This is appropriate to present the user with a choice of options i.e. navigation, and multiple choice options.
Source:	Literature Survey: 2.6.2

Requirement:	29
Description:	The system should make judicious use of form fillin
Rationale:	This is not necessarily appropriate given the experience of some of the users, but may be necessary to obtain user information, or in higher levels.
Source:	Literature Survey: 2.6.1, 2.6.2

Requirement:	30
Description:	The system shall use tool tips or labels on buttons
Rationale:	So that the user knows the purpose of the button.
Source:	System Evaluations: Education City, Primary Games

Requirement:	31
Description:	The system shall use high contrast colour combinations
Rationale:	Information may not be obvious enough if there is not a high enough figure-ground contrast. Also if it is to be used on an interactive whiteboard, these often suffer poor visibility when positioned in direct sunlight. The positioning of the interactive whiteboard cannot be controlled; however, high contrast colour combinations may be more visible in these conditions.
Source:	Stakeholders: Year 3 Mathematics lesson observation System Evaluations: Education City, Primary Games

Requirement:	32
Description:	The system should not use colours that people with colour blindness may have difficulty in distinguishing to indicate meaning
Rationale:	Red and green may not be distinguishable, so should not be used to indicate if an answer is right or wrong.
Source:	System Evaluations: Education City

Requirement:	33
Description:	The system should use large sans serif fonts that can be read from a reasonable distance from the display screen
Rationale:	Reading text on a large screen, such as an interactive whiteboard, takes time and effort. Users may be located at various distances from the display, so the text must be of a sufficient size and style to be readable from a reasonable distance away from the screen.
Source:	Stakeholders: Year 3 Mathematics lesson observation, year 4 ICT lesson observation Literature Survey: 2.5.1

Requirement:	34
Description:	The system should not require the user to interact with the top half of the interface
Rationale:	If the system is used on an interactive whiteboard, the top part of the interface may be inaccessible to some users who cannot reach.
Source:	Stakeholders: Year 3 mathematics lesson observation, interview year 2 child

Requirement:	35
Description:	The system shall not require the user to 'write' with the pointing device
Rationale:	Text written with an electronic pointing device is difficult to read and unrecognizable to the computer.
Source:	Literature Survey: 2.4.1

Requirement:	36
Description:	The system should minimise typed input
Rationale:	Typing can be seen as a daunting task for most children which can take over the direction of the whole exercise.
Source:	Literature Survey: 2.5.2

Requirement:	37
Description:	The system shall not use verbal audio output
Rationale:	Recorded voices are not always very clear and can be difficult for children to understand. In a classroom environment, there may be too much background noise for the system noises to be audible. Audio is distracting and can be detrimental to others in the setting.
Source:	Stakeholders: Year 4 Mathematics lesson observation Literature Survey: 2.5.1

Requirement:	38
Description:	The system should only display functionality necessary to complete a task
Rationale:	Extra functionality that is not needed to complete the task may cause confusion. The user may accidentally do something not intended. Context menus in applications can be distracting or disruptive.
Source:	Stakeholders: Year 4 ICT lesson observation System Evaluations: Education City

Requirement:	39
Description:	The system should meet Shneiderman's (1998) 'Eight Golden Rules of Interface Design' (See Appendix A)
Rationale:	These principles are considered to be applicable in most interactive systems, so should be followed.
Source:	Literature Survey: 2.5

3.4 Conclusion

The requirements have been sourced from a combination of the literature survey, stakeholder involvement and evaluation of existing systems. The main difficulty with sourcing from the literature was the extent of information available, particularly relating to usability guidelines. It was at times difficult to select which guidelines were applicable and complementary. One area of conflict relates to the use of audio output from the system. The literature indicates that learning is easier when information is presented visually and verbally. However, audio output may distract others in the environment not using the system. Headphones could be used to resolve this; however, this is not suitable if the system is being used by a pair or group. One of the teacher interviews also identified that children can have difficulty understanding recorded voices. For these reasons it is required that audio output is avoided.

The involvement of stakeholders was useful. However, stakeholder involvement was limited with only two teachers and two children interviewed. Also, only lessons in years 3 and year 4 were observed. It would have been valuable to observe the higher two years of key stage two as teaching is likely to be quite different. Clearly, interviews and observations are highly subjective, meaning that it is likely that some important information may have been missed and other information given higher precedence than was appropriate. With this in mind, the validity of the findings from this stakeholder involvement cannot necessarily be known. However, as lesson observations were backed up by teacher interviews, the findings from these are likely to hold more weight than if each method had been used in isolation.

With the system evaluations, many usability issues were identified with suggestions made for improvement, some of which have been carried forward to the requirements specification. Positive aspects were identified less often so there have been fewer opportunities to use 'good' aspects in the requirements. There would also have been value in evaluating these systems with children as an expert cannot anticipate every problem a child will have. Also, the systems should have been evaluated on interactive whiteboards but this was not possible.

There was some difficulty specifying the requirements relating to interactive whiteboards. For the system to function on an interactive whiteboard it must also function on a PC because interactive whiteboards are used though PCs. Also, it cannot be guaranteed that the system will always be used on an interactive whiteboard, meaning that the system should be usable on both PCs and interactive whiteboards. Clearly, there are differences in the way these are used; therefore, there may be conflict in making the system usable for both types of medium.

A further area of difficulty was in specifying the requirements to support collaboration. The scope has been restricted so that the system need only support collaboration implicitly by providing learning content that is challenging enough to encourage collaboration. Clearly, this is difficult to specify as it cannot necessarily be known what an appropriate level of challenge is, particularly when the system may be used by a group of mixed ability children.

Another area of difficulty relates to requirements that rely on subjective use. The system is required to provide a suitably enjoyable and challenging learning experience through use of concrete and familiar concepts with familiar vocabulary. These are necessary if system is to benefit learning, however, each user is different and what is suitable and familiar to one may not be to another. This indicates a need for further user involvement throughout the design process. In view of this the next chapter will detail the user-centred methods used in design.

4 Design Methods

4.1 Introduction

This chapter will discuss the methods used to design a suitable system to meet the requirements. Ultimately the resulting system should be suitable and highly usable for the intended user group by taking into account their needs, capabilities and preferences. To achieve this, it is necessary to continue user and stakeholder involvement during the design process, by running participatory design sessions. From this it should be possible to move towards a design specification for a prototype system (see Chapter 5).

4.2 Participatory Design

The participatory design methods of brainstorming, CARD and PICTIVE will be used with a group of children from Moorlands Junior School, who will now be referred to as the ‘Design Team’. The main purpose of these sessions will be to generate and try out initial ideas for the system. As discussed in the literature survey (see Section 2.7), where children are involved in design, the approach should be modified to reflect their ability and understanding, with the adult in the group leading the processes and discussions. To help plan for these sessions, a pilot brainstorming session will be carried out. Key findings from this will then be used to suitably modify the methods used in the subsequent participatory design sessions.

4.2.1 Pilot Brainstorm (26-Nov-2006)

The pilot session was carried out with the two sisters, aged seven and nine, who were previously interviewed. The materials used were large pieces of paper and a number of different coloured pens. The children were asked to talk about ideas for software to help them learn maths in a fun way and possibly whilst working with other children. The ideas were recorded by the children in the form of a large spider diagram (see Appendix F). Interesting ideas from this were:

- A ‘snakes and ladders’ game with number questions on each square. The winner’s prize would be a video about numbers with a real person, puppet or robot talking
- A game where a picture is revealed by answering maths questions
- A running race game where questions have to be answered to move around the track

The children’s ideas were mostly based on existing games, rather than innovative ideas. It was felt that they did not fully understand what was asked of them as most ideas were competitive rather than co-operative. After these initial ideas, the children appeared to start losing interest and attention, so it was decided to change the approach. The children were asked about their interests and favourite things instead in order to get a general idea of what girls of that age like. A new spider diagram was used to capture these things, which included:

- Computers, Nintendo DS, Tamagotchi (a virtual pet toy)
- Puppies, kitten, dolphins, penguins

From this it can be seen that the children have an interest in technology and animals, however, it is not known if this is common across the primary stakeholder group. This should be investigated further because there could be potential to use animals as a theme for the system. Having an animal theme is further backed up by the mention of animals in terms

of relating work to what children know about and are interested in the interview with the year 4 teacher. Furthermore, animals could be used as a concrete concept to base learning around.

As the discussion around interests and favourite things was successful in the pilot, this will also be done at Moorlands Junior School in a brainstorming session. This is likely to keep the children's attention and will allow stakeholder interests to be investigated further. A particular focus will be made on animals to establish if this is actually a common area of interest and whether the children would like to play a maths game with an animal theme.

A further finding from the pilot that can be applied to the subsequent participatory design sessions is the need for an audio recording to be made of the session. The pilot was not audio recorded, however, a recording would have been valuable as it was difficult to capture what the children said whilst running the session. Although the children wrote down some ideas on the spider diagram, interpreting this was difficult because of the children's handwriting.

By recording future participatory design sessions, the primary focus can be on interacting with the children as to make the sessions more effective by keeping the children interested and involved. Analysis can take place by listening to the recordings and if there is any ambiguity in the interpretation, the children can be questioned in the next session to clarify meaning. In this way less information will be lost and the data should be more accurate.

4.2.2 Design Team Brainstorm (19-Jan-2006)

Four 'mixed-ability' year 5 children, aged nine and ten, were involved in this first participatory design session at Moorlands Junior School; two female (Child A and Child B) and two male (Child D and Child D). The children were provided with coloured pens and large pieces of paper. They were told to say their ideas and write them down as a spider diagram whilst listening to each other for inspiration (see Appendix G for the spider diagrams). Prompts in the form of keywords were used to investigate the children's interests. The prompts and interesting findings include:

Prompt: Favourite hobby, game and toy

- Three children mentioned sports, the fourth said art as their favourite hobby
- Child A has a Nintendo DS and likes to play 'Nintendo Dogs'
- Child E has a PlayStation Two and likes to play 'James Bond'
- Some children like to play board games, including 'Monopoly' and 'Cluedo'

Prompt: Favourite animal

- The children liked talking about animals; their favourite animals include tigers, parrots, dolphins, dogs, cats and polar bears
- The children all thought a computer game to do with animals would interest them; but they were not sure if it would be more interesting than a sports game

Prompt: Favourite place

- Favourite places included places the children have been to, such as Florida, Devon and Cornwall and others they would like to go to, such as New York and Australia

Prompt: Favourite colour

- The children thought that most children like bright colours

Prompt: Favourite and least favourite maths topics

- There were a range of views about maths, some positive and some negative
- The children were of different mathematical abilities, so each had their own opinions about maths and the topics covered, as well as what is 'easy' and 'difficult'
- All but Child E thought division is difficult

There seemed to be two common areas of interest with the design team children, these were sports and animals. In view of this, these areas were taken forward as the session moved on to generating ideas for the system. The children were told the software would be used by children in years 3 to 6 to help them learn, possibly in a group. Although the requirements scope (see Section 3.3.2) limits the prototype to a single-player game, it was still necessary to investigate the children's views on how a multi-player game could be done. The design team were also told the software could be a game, that it might be used on an interactive whiteboard and that it should be fun and enjoyable to use. The children were then asked what the theme for the game could be:

Prompt: Theme

- A darts game with the computer throwing the darts. The player has to add up the points. The player would do division and multiplication to make it more difficult
- A racing game where the player adds up the speeds of the cars or there could be cones in the road with questions at each cone that have to be answered to move on
- A game with different types of animals that are won by answering maths questions

As the session was only one hour long, it was necessary to choose one idea to take forward as there would not be time to explore each idea fully. Due to the previous thoughts on an animal theme, this was the one chosen to continue with and the children were happy with this decision. The animal theme was then expanded to that of a game to build a zoo. To elicit information about different aspects of this game, prompts in the form of keywords were used. The prompts and interesting findings include:

Prompt: Characters

- The player could play a character that has been hired to build the zoo
- In a multi-player game, there could be other characters such as the zoo manager or zoo keepers for different types of animals

Prompt: Testing and questions

- There should be animal related questions; Child D said "If you want a racoon in the zoo, you have to answer a question about it", Child D thought that money could also be won
- The questions should be different depending on the type of animal, child E said "It would be harder to get rarer animals"
- There should be different levels with questions increasing in difficulty. Child E thought the last level would have to be quite special, he said "they always make the last level quite hard because you're very close to finishing"
- If the player gets a question wrong, Child E thought that points should be lost and Child D said "every time you lose a question an animal goes out"
- There should be no time limits when answering questions, Child D said "a time limit would put you under pressure"

- The player should be able to choose the game difficulty, based on school year and ability. Child E said “you [the player] could pick hard ones or easy ones”
- The questions should change, Child E said “You [the game] should change it, they should be random, it would be luck if you [the player] got the same ones again”

Prompt: Help and tutoring

- To get help, Child E said “you [the player] could go and visit sort of, another person who runs it and ask them what to do” possibly at an information point
- Child B thought help should give “a strategy to help you [the player] know what sort of thing to do to work it out”. Child D thought the help should use diagrams in explanations

Prompt: Saving

- It should be possible to save a game
- Opening a saved game will allow the player to continue from where they saved

Prompt: Winning

- The game will be finished when the zoo is complete, Child D said “yeah it [the game] says you have safely made your zoo, congratulations”
- Child E thought the prize could be the address of a website about an animal of the player’s choice. Child D thought the prize could be a picture of the player’s favourite animal which could be printed out and coloured in

4.2.3 Design Team CARD (03-Feb-2006)

The brainstorming session resulted in the initial idea for a ‘Maths Zoo’ game. This idea was developed further by utilising other participatory design methods in the subsequent sessions with the children, the first of which being CARD. Four children were involved in the CARD session, this included Child B and Child D. However, Child A and Child E were unavailable, so two new children were involved, Child C (female) and Child F (male), also from year 5 at Moorlands Junior School.

The group were provided with a set of coloured index cards, coloured pens and a large piece of paper. Muller (2001, p.90) advocates using coloured cards to ‘indicate a high-level class hierarchy, with related cards printed in the same color.’ Based on Muller’s (2001) work on Layered Participatory Analysis, it was decided that three types of CARD would be used:

- 1. Observable and Formal components** (yellow cards) - Work objects and activities that would be visible to an outside observer
- 2. Skill and craft** (pink cards) - Work that might not be apparent to an outside observer. Including knowledgeable practice and purely internal activities, such as setting goals, developing strategies, making plans, and establishing motivations
- 3. Description** (green cards) - Additional tools for interpreting the concepts covered in the preceding two categories

The current literature about the CARD method does not go into much detail about what data should be captured on the cards; however, there are two examples in Muller’s (2001) paper. From this and by considering what information would be useful, it was decided that the following fields would be used to capture information on the cards:

- **Title** - brief summary of the task/activity
- **Description** - detailed description of the task/activity
- **Purpose** - intention of the task/activity
- **Input** - required inputs
- **Role(s)** - role or roles involved in the task/activity
- **Output** - expected outputs

The session started by recapping on the ideas for the Maths Zoo game. The children were told the aim of the session was to create cards and arrange them to detail how the Maths Zoo game would be played. The different types of cards were then shown and explained to them. As with the previous brainstorming session, prompts in the form of keywords were used for guidance. However, these prompts were aimed at exploring a wide range of possible scenarios and aspects of system functionality. The prompts and interesting findings include:

Prompt: Single-player versus multi-player game

- At the start of the game, a single-player or multi-player game can be chosen
- In a single-player game, the player will be the zoo keeper. They will answer questions but other computer players may 'help' them with the questions
- In a multi-player game, each player will be a zoo keeper for different types of animals. Questions will have sub-parts, with each zoo keeper required to answer their part in order to complete the whole question

Prompt: Game Layout

- The zoo will have different zones for different types of animals. The animals will ask the questions and the questions will be related to the animals in that zone
- There will be a map of the zoo showing the different zones in the zoo
- There will be a coloured path through the zoo, with colour used to indicate progress. Green will show that a zone has been completed; yellow will show that a zone is in progress and red will show that a zone still needs to be completed

Prompt: Starting a new game and resuming a saved game

- For a new game, there will be the option to choose school year from those in key stage two
- There will be the option to choose game difficulty from easy, medium or hard
- The player will have to give each new game a unique name and password, for saving purposes. There will be an onscreen keyboard to enter this on an interactive whiteboard
- To load a saved game, a selection of previously saved games can be chosen from

Prompt: Answering questions correctly

- The reward for a correct answer will be an animal and points/money
- A male and female of each type of animal needs to be won in each zone
- The prize for completing the game will be a certificate with a picture of an animal from a choice of animals, that can be printed out and coloured in

Prompt: Answering questions incorrectly

- Incorrect answers will result in points/money being deducted
- When there are no points/money left, the most recently won animal will be deducted
- If there are no more animals left, the player can do a bonus question to win money in order to keep playing the game. If the bonus question is answered incorrectly, the game will be lost and the player will have to start a new game

Prompt: Seeking help

- If help is needed the player can go to the zoo keeper's hut
- The help will be tailored to match the point in the game, in order to give appropriate help
- The help will be related to the game difficulty and the question's mathematical content rather than game play and interaction
- There will be instructions that can be accessed from any point in the game. These instructions will provide game play and interaction help

The ordered cards from this session can be seen in Appendix H.

4.2.4 Design Team PICTIVE (06-Feb-2006)

PICTIVE was used with the intention of developing a low-fidelity paper prototype of the Maths Zoo (See Appendix I for the resulting designs). Four children were involved in the PICTIVE session; these were Children A, B, C and D, each of which had been involved in at least one previous session. With PICTIVE, basic stationery is used to design interactively on a shared design surface. The children were provided with large sheets of paper, a variety of coloured pens, paper and sticky notes. To guide the one hour session, it was necessary to plan what would be discussed for inclusion in the paper prototype, so a number of prompts in the form of keywords were prepared. The prompts and interesting findings include:

Prompt: Player characters

- Each child drew what they thought the zoo keeper should look like
- The girls all drew female zoo keepers and the boy drew a male zoo keeper
- The zoo keepers were all the same age as the children
- The children thought they should be able to choose what zoo keeper to play as

Prompt: Instructions and help

- Each child drew an icon for 'Help' on a sticky note. One idea was a question mark, another was a person thinking
- Child A thought the help button should go at the top of the screen, because she thought it would not be seen at the bottom. However, she changed her mind and said it should go at the side after it was explained that the top half of the screen may be unreachable on an interactive whiteboard
- It was explained that buttons might still be unreachable if positioned past a certain point on the side. Child D suggested that buttons could stop half way up the screen
- The other children thought the buttons should go on the bottom of the screen
- Each child drew an icon for 'Instructions' in a sticky note. Child D drew a big 'I', as this is the first letter of the word 'instructions'; another idea was a leaflet icon because instructions usually come in leaflets

Prompt: Other icons

- The children were asked if they could think of any other things that would need to have icons. Child C thought 'Tools' might be needed for printing
- Child D thought a 'Quit' button would be needed. Icons drawn for this included a person leaving a room through a door and a large cross
- The children also thought a 'Save' button may be needed. Child A drew a file with paper going into it, and Child D drew a memory card with an arrow pointing into it
- Child D thought a 'Zoom' button might be needed to zoom in and out of the map
- The children were asked if they would want to see their game status, for example their score or details about the level they are on. For this Child D drew a pie chart
- The sticky notes representing the buttons were positioned along the bottom of the paper representing the screen. These were then rearranged into a logical and practical order with only the most appropriate button of each type visible

Prompt: Navigation and the map

- Child A thought the map should have areas with different families of animals and that there should be a winding path leading between the areas. The map was drawn on the paper representing the screen and the path was drawn to wind around the 'screen' with animal areas positioned around the path
- Child D suggested having a Café, where questions would be related to food. This area was also drawn onto the map
- The children drew different types of animals on sticky notes then positioned these in different areas, with similar animals together
- Child D suggested having a car to move between the areas. Child A thought the car could be stopped at the areas by road items such as cones, lollipops and traffic lights. These were drawn on sticky notes then positioned on the map

Prompt: Questions

- A new sheet of paper was used to draw the question screen. Child A drew a large picture of a rabbit on the left hand side of the screen with a speech bubble coming out of the rabbit's mouth asking the question
- The children were asked how an answer would be selected or entered. One idea was to have multiple choice answer buttons, so an example of this was drawn
- Another idea was to use check boxes. The children were told that check boxes can be difficult to use because of their size and would need a submit button. Child D did not understand what 'submit' meant, so a different word may have to be used
- Child D thought the player could type the answer in a box, making it harder
- Child D suggested having a menu list with answers; however, it was decided that the player should be able to see all answer choices at once
- It was decided that the way in which the answer is entered should depend on the level in the game. In lower levels, multiple choice buttons could be used and at higher levels, the player will have to enter the answer freely in an answer box

Prompt: Winning the game

- A new sheet of paper was used to draw a certificate with the players name and “well done” written on it, as suggested by Child C
- Child A suggested having a medal on the certificate, so this was drawn
- It was decided that the certificate should have a picture of the player’s favourite animal which can be coloured in when the certificate is printed, so an animal was drawn

The resulting artefacts from the PICTIVE session can be considered as a low-fidelity paper prototype (see Appendix I). However, there are other ways of prototyping and these will now be considered for use in the project.

4.3 Prototyping

According to Preece et al., ‘A prototype is a limited representation of a design that allows users to interact with it and to explore its suitability’ (2002, p.241). A prototype can be considered as low-fidelity if ‘it does not look very much like the final product’. They often use ‘materials that are very different from the intended final version, such as paper’ (Preece et al. 2002, p.243). Low-fidelity prototypes are simple, cheap and quick to produce, thus they allow ideas to be explored readily.

From the low-fidelity prototype and the ideas generated in the participatory design, the next step is to iteratively design and develop a high-fidelity Maths Zoo prototype. According to Preece et al. (2002, p.245) ‘high-fidelity prototyping uses materials that you would expect to be in the final product’ and look ‘much more like the final thing’. High-fidelity prototypes are often software based. As stated by Rudd et al. (1996) they are fully interactive, meaning the user interacts with it as if it were a real product. To achieve this level of interaction, a substantial programming effort is usually needed, thus making the process of high-fidelity prototyping lengthier and more demanding than low-fidelity prototyping.

It is common for high-fidelity prototypes to have limited functionality and scope. This can be done in two ways, either by developing horizontal or vertical prototypes. Horizontal prototypes have a wide range of functions but with little detail. Vertical prototype have great depth, but for a small range of functions. A vertical high-fidelity prototype will be developed and will be the resulting implementation of the system. To achieve this, the prototype will be evaluated iteratively by stakeholders and the feedback and findings used to improve the design of the system in the each iteration of the prototype (see Chapter 6).

4.4 Conclusion

This chapter has presented an overview of the activities and findings involved in the user-centred design process which has resulted in the low-fidelity prototype of the ‘Maths Zoo’ system. The design process began with a pilot brainstorming session to gather initial ideas and help guide the subsequent participatory design sessions in Moorlands Junior School. From this pilot, the need for directed brainstorming was identified, as to make the best use of the time and keep the children interested. Furthermore, the need for audio recording was identified otherwise valuable information would be lost and interaction hindered.

Brainstorming was then used in Moorlands Junior School with a group of four children, known as the design team. During this session, several themes for a system were discussed, including themes relating to sports and animals. Based on previous stakeholder involvement it was known that children often have an interest in animals and that this is familiar to them. For this reason, the animal theme was pursued and by the end of this session the idea for a 'Maths Zoo' game had emerged where the player answers maths questions to win animals.

This brainstorming was successful as it resulted in a suitable and viable idea for the system. However, the children seemed uncomfortable writing down their ideas and had to be prompted to do so. This may have been because they found it difficult reaching over the large table to get to the paper or because some children needed spelling help. However, as the session was recorded, having the ideas written down was not essential. It was also noticed that the boys were more vocal and forthcoming with their ideas; this may have been because they had experience playing computer games and drew on this experience for new ideas.

Following the brainstorming session, CARD was used with the design team. In this session the idea for the 'Maths Zoo' game was expanded to get an idea of how the game would be played by ordering cards that related to different aspects of the game. The session was to some extent successful because valuable information was obtained, however, this was mostly from the audio recording rather than from the cards. During the CARD session the children did not seem comfortable writing on the cards, therefore they spoke their ideas and these were written down for them. It seemed that the children did not really understand the difference between the three types of cards, therefore, only the Observable and Formal component cards were used. However, they did get involved with ordering the cards on a large sheet of paper and drawing arrows between them to indicate the flow of tasks.

As the CARD session was only an hour long it was difficult to go into the detail required for each card, so the planned structure of the cards was not kept to. Instead, it was easier to capture the high-level task at the top of the card, with relevant detail listed underneath. For example, one card had a high-level task of 'Start Game' and then the detail included choosing the number of players and choosing the difficulty from easy, medium and hard.

The final session with the design team used a variation on PICTIVE, as it was audio recorded rather than video recorded. The children really enjoy this session because it was more 'hands-on' than previous ones. The session provided a great deal of information, from both the drawings and audio recording. However, as the session was only an hour long it was not possible to cover every aspect of the system, including mathematical content.

As the participatory design sessions have not allowed every feature and aspect of the system to be designed, it will be necessary to make some design decisions without input from the design team. These decisions will be discussed in the next chapter as the design becomes formalised using the information and artefacts generated in these participatory design sessions as a basis. It will then be necessary to involve stakeholders again in iterative evaluation and re-design cycles to ensure the prototype system meets their needs and expectations (see Chapter 6).

5 User Interface and System Design

5.1 Introduction

Based on the ideas and low-fidelity prototype as detailed in the previous chapter, this chapter will capture the system's functionality, behaviour and interface starting at a high-level and then moving to more detailed design. A basic high-fidelity prototype, known as Maths Zoo Alpha (see Appendix K), will then be implemented based on this design. This prototype will then go through two phases of user evaluations and re-designs, as detailed in the next chapter.

5.2 High-Level Design Decisions

As the system will be implemented as a high-fidelity vertical prototype, only a sub-set of functionality will be implemented, but to a detailed level. The scope of this will now be defined in terms of the year group, mathematical ability (see Figure 5.1) and mathematical content the system will cover.

The system is intended to be used by children in key stage two, which encompasses four school years. To provide content for each of these years (Requirement 21), a significant number of questions and activities would be needed. It has been decided that the scope of this should be limited to a single year group. As the design group children were all from year 5 (aged nine and ten, it seems appropriate to cater for this year.

Within each year group are children of different mathematical abilities, usually organised into sets. To cover each set, different questions and activities would be needed to match the abilities and objectives of the sets (Requirement 22). The scope of this will be limited to the middle ability set as this is likely to be relevant to the majority of year 5 children. Also, the teachers interviewed at Moorlands Junior School said they usually aim examples and questions at the middle ability set.

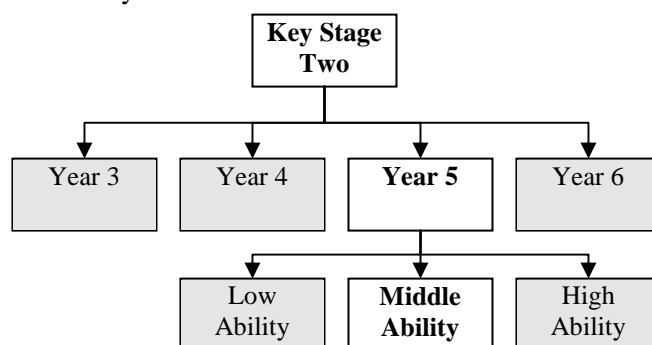


Figure 5.1

The system is required to provide mathematical instruction linked to the mathematics national curriculum (Requirement 1). Clearly to achieve this, a significant amount of content would be needed, which is not appropriate given the nature of the prototype. Therefore, the scope of this will be limited so that the system only covers a specific mathematical topic. From the analysis with stakeholders it is known that children in year 5 often have difficulty grasping the concept of division, so this is the topic that the system will focus on. The will be achieved by dedicating one animal area (zone) in the zoo to division.

Furthermore, as only one zone will be implemented, it is not necessary to implement the back-end of the saving functionality. This is because there are not enough tasks for the user to do that would justify the need for save functionality in the prototype. However, the interface for this should be put in place so it can be seen how this would be implemented in a full version of the Maths Zoo game.

5.3 System Architecture

It is necessary to establish a basic structure and framework for the Maths Zoo system in terms of the sub-systems and major components and the ways in which these will interact and communicate. These aspects are modelled in a high-level block diagram (see Figure 5.2). Each block represents a sub-system, with some divided further as shown where there are boxes within boxes. Arrows indicate the flow and direction of data and/or control between the sub-systems. This type of diagram does not make a clear distinction between the flows of data or control, so each of the sub-systems, components and interactions between them will be discussed in Table 5.1.

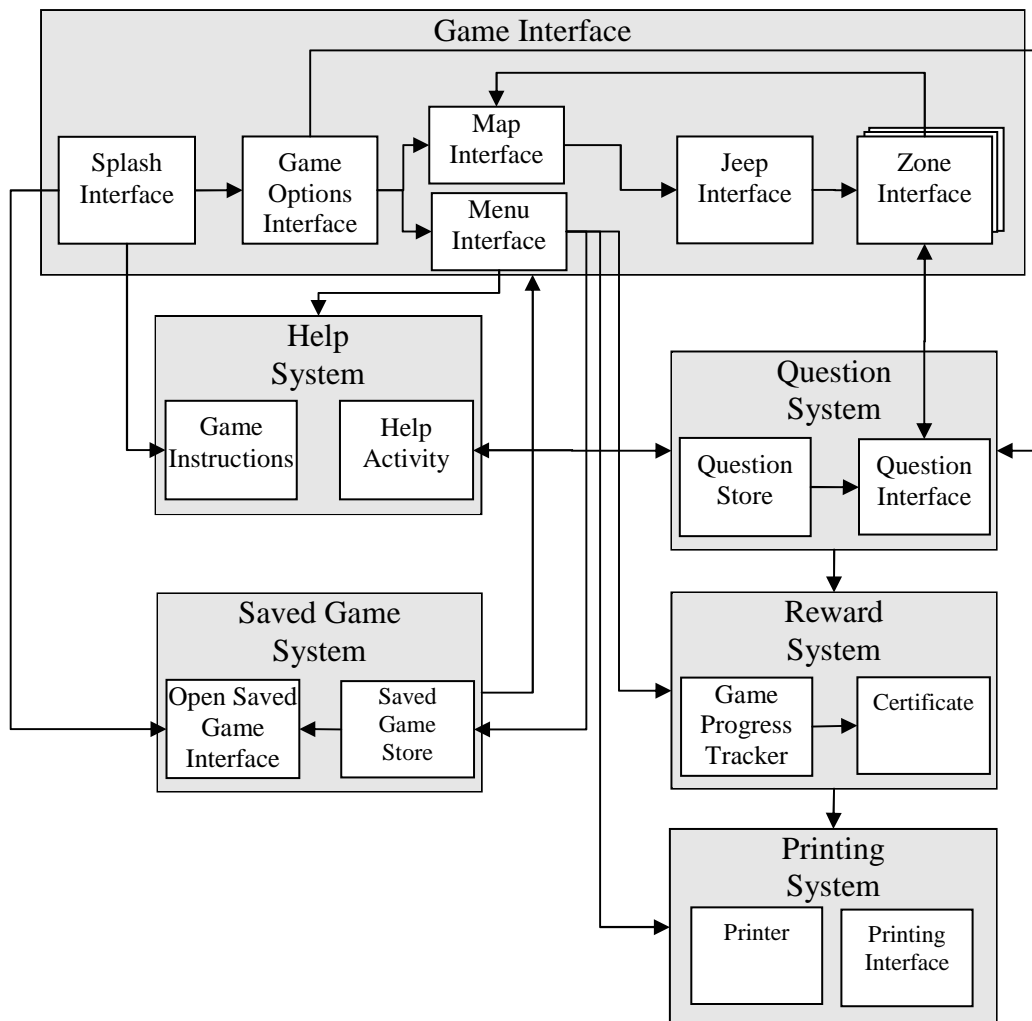


Figure 5.2 Architectural block diagram for the maths zoo system

Table 5.1 System Sub-Systems, Components and Interactions in the Maths Zoo system

Sub-System	Component	Description	Interactions
Game Interface	Splash interface	User can access game instructions, start a new game or open a saved game	<ul style="list-style-type: none"> • Links to game instructions • Links to the game options interface • Links to the open saved game interface
	Game options interface	User interacts to choose game options to tailor a new game	<ul style="list-style-type: none"> • The game options chosen influence which questions are used from question database • The game options chosen will influence which zoo keeper is used in the game
	Map interface	Shows each zone and indicates progress made in completing each zone in the zoo	<ul style="list-style-type: none"> • Indicates zone progress based on game progress tracker data • Links to the jeep interface when a zone has been selected, which then links to the selected zone
	Menu interface	Area of user interaction, options available dependent on the current interface	<ul style="list-style-type: none"> • Links to the help system • Links to the map interface • Links to save the game • Links to print the certificate • Links to exit Maths Zoo
	Jeep interface	Shows movement to a specific zone	<ul style="list-style-type: none"> • Links to the specific zone that has been selected from the map
	Zone interface	Shows progress in completing zone by pictographic representations of animals; route to Question system	<ul style="list-style-type: none"> • Indicates the animals that have been won and the animals that need to be won in the zone, based on the data stored in the game progress tracker • Links to questions in order to win animals for the zone
Question System	Question interface	Displays question and animal	<ul style="list-style-type: none"> • Question data from the question store
	Question store	Store of questions organised by year and difficulty	<ul style="list-style-type: none"> • Input to question interface dependent on the game options selected at the start of the game
Help System	Game Instructions	How to play Maths Zoo	<ul style="list-style-type: none"> • Linked to from the menu • Links back to previous sub-system
	Help Activity	Activity to help with a specific question and topic	<ul style="list-style-type: none"> • Linked to from question interface • Links back to question interface • Help tailored to current question
Reward System	Game progress tracker	Stores which animals have been won	<ul style="list-style-type: none"> • Updated when a question is answered correctly or incorrectly • Data used by map and interface to indicate zone progress
	Certificate	Certificate can be personalised on completion of a zone	<ul style="list-style-type: none"> • Game progress tracker used to verify zone is complete and get zone score • Certificate can be printed, if system is connected to a printer

Printer System	Printer	External printer system	• Certificate printing
	Printer Interface	Interface to send star printing	• Sending certificate for printing at the printer
Saved Game Store	Open saved game Interface	Interface to open a previously saved game	• This will not be implemented due to high level design decisions, but it would allow game data retrieval
	Saved game store	Store of saved games	• This will not be implemented due to high level design decisions, but it would allow game data to be saved

5.4 Detailed Design

The detailed design will be structured around the system components identified in Table 5.1.

5.4.1 Splash interface

The splash interface will feature the Maths Zoo logo (see Figure 5.3) and have button linking to the instructions, to start a new game and to open and saved game. The logo has been designed without input from the children but because it does not form a key part of functionality, user involvement was not essential.

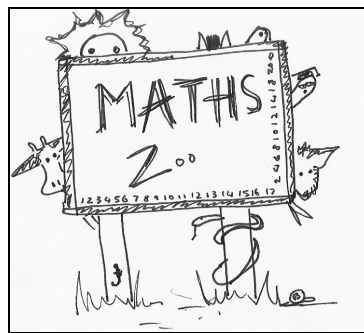


Figure 5.3 The Maths Zoo logo

5.4.2 Game options interface

The game options interface will have buttons to choose the number of players, school year and game difficulty in tabular format (Figure 5.4). Only 'One' player, year '5' and 'Medium' difficulty will be available; the other will be greyed out to indicate unavailability. When an option is selected this will be made obvious by the appearance of a box around the option. The system will check all options have been selected before allowing the player to continue.

Number of Players	<div>One</div>		Two	
School Year	3	4	5	6
Difficulty	Easy	Medium	Hard	

Figure 5.4 Game options table

Also as part of starting a new game, the Zoo Keeper persona will need to be selected. Four zoo keepers have been designed (see Figure 5.5) based on those drawn by the design team in the PICTIVE session (see Appendix I). Each zoo keeper is in the same age group as children in key stage two. The player will choose a zoo keeper by directly clicking on the picture of one. This selection will be obvious by the appearance of a box around the chosen zoo keeper. The system will check the player has selected a zoo keeper before the player can continue.

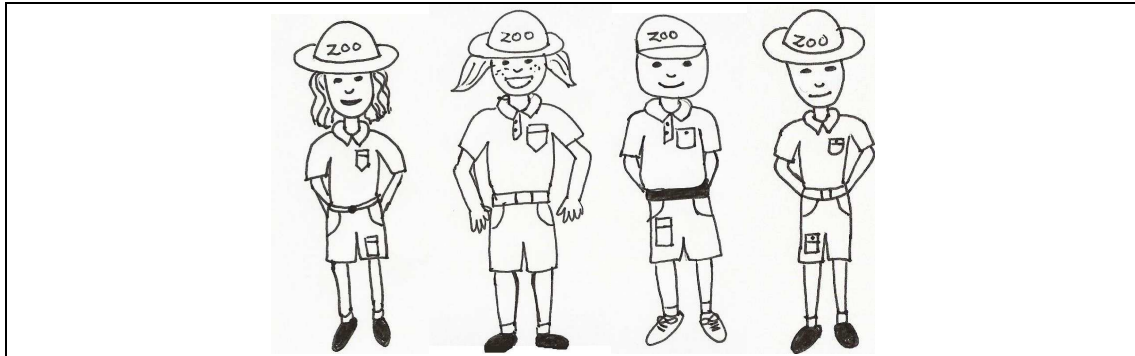


Figure 5.5 The zoo keepers

5.4.3 Map interface

The map interface will show each zone positioned around the zoo track (see Figure 5.6), as based on the design from the PICTIVE design session (See Appendix I). Each zone on the map will be a button linking to the zone.

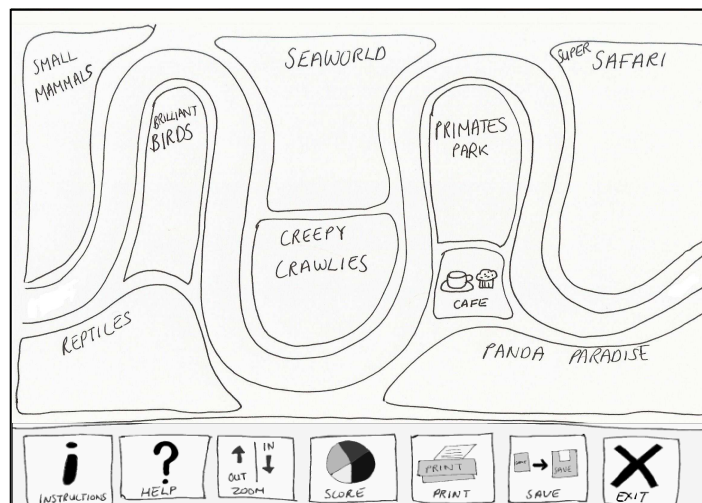


Figure 5.6 The map and menu interface

5.4.4 Menu interface

The menu will be located in a bar along the bottom of the screen (see Figure 5.6). The menu will consist of a series of buttons as designed in the PICTIVE design session (see Appendix I). However, the 'Zoom' button is no longer needed as the whole map should be seen at once and a 'Score' button is not needed because the score should be visible at all times. The buttons shown on the menu will be tailored to the rest of the screen i.e. 'Print' will only be available when they certificate is being viewed.

5.4.5 Jeep interface

The jeep interface is used to move from the map to a zone, it will consist of an animated jeep moving across the screen until it is stopped by a road item (See Figure 5.7). The interface will have a 'Next' button to give the user control over when to move on to the zone. The design team had originally wanted to see a jeep move along the path on the map, however, it was felt the jeep would not be seen clearly on the map.

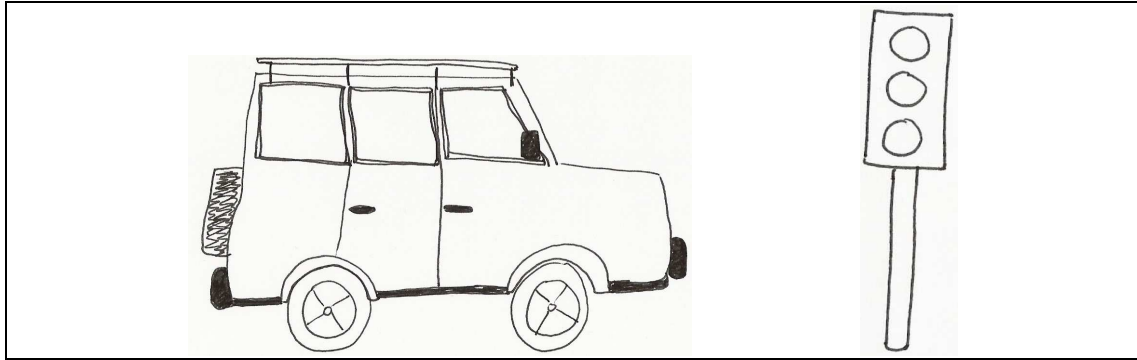



Figure 5.7 The jeep and a road item

5.4.6 Zone interface

Only the 'Super Safari Zone' zone will be implemented (see Figure 5.8), it will have giraffes, elephants and lions. The zone will have a greeting: "Welcome to the Super Safari Zone" and a picture of the zone so that the player knows what zone they are in. The zone will inform the player that they need to win a male and female of each type of animal by answering maths questions. To show the progress in with this, a table will be used with each cell representing an animal. Where the animal has been won, the animal will be shown, otherwise there will be a button linking to a question.

Welcome to the Super Safari Zone

Try and win a male and female of each type of animal by answering maths questions




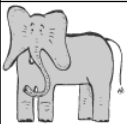

	Male	Female
Giraffe		Get Animal
Elephant	Get Animal	
Lion		Get Animal

Figure 5.8 The Super Safari Zone interface

5.4.7 Question interface

The questions interface will show the animal the player is trying to win and a question with four multiple choice answer buttons. A generic question layout can be seen in Figure 5.9. If the player clicks the correct answer, they will be taken to another page and informed that the animal has been won and the table in the zone will then be updated. If the player clicks on a wrong answer, they will be taken to another page and informed that the question was incorrect, and then can try again once more.

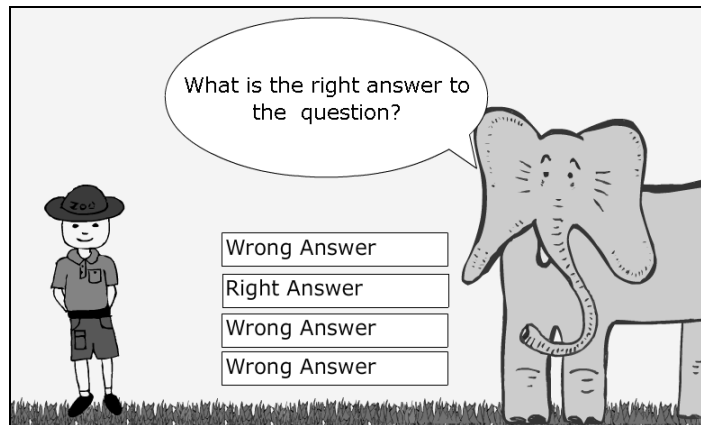


Figure 5.9 The question interface for a generic question

5.4.8 Question store

The questions had not been designed during the participatory design sessions, so it was necessary to design these without input from the design team. The questions will mostly be based on division, with some use of multiplication as these two topics are commonly taught together. The questions will make use of zoo related concepts, for example, animals, their food and living environments within the zoo; thus making the mathematics ‘concrete’. Initially, one question will be implemented for each animal then based on evaluation feedback further will be developed. The initial questions, with the multiple choice and correct answers can be found in Table 5.2.

Table 5.2 Question design

Animal	Question	Multiple Choice	Correct
Giraffe	There are 12 giraffes but only 3 giraffe enclosures. The zoo keeper wants to put equal numbers of giraffes into each enclosure. How many giraffes should go in each enclosure?	2, 3, 4, 5	4
Elephant	Every day the zoo keeper needs to make sure there is enough food for the elephants. There are 6 elephants, each needs to each 3 buckets of food every day. How many buckets of food are needed every day for the elephants?	6, 12, 18, 24	18
Lion	There is £3000 available to build 10 new lion huts, what is the budget for each lion hut?	£3, £30, £300, £3000	£300

5.4.9 Game progress tracker

The game progress tracker will store which of the animals in a zone have been won.

5.4.10 Game Instructions

The instructions will include information on the aim of the game and how to play the game including information about the map and the zones; winning animals; accessing and using help activities; completing zones and winning a certificate. See Appendix J for the full text of these instructions.

5.4.11 Help Activity

Help in the system will only relate to learning and education help as interaction and navigation help is not necessary given that the game is intended to be simple to use. Furthermore, the instructions will provide guidance on how to play the game and extra advice will feature directly on the interface where appropriate, for example, 'hints' on help activities.

Each question will have an associated help activity, which provides the player with appropriate 'scaffolding' to help work out the question. On the help activity, the question will be shown again and a hint will be given telling the player what they need to do. Most help activities will allow the player to directly manipulate the concrete objects as mentioned in the question through use of drag and drop.

Giraffe Help Activity

Based on the giraffe question (see Table 5.2), the giraffe help activity (see Figure 5.10) will feature 12 giraffes and three giraffe enclosures. The player will be able to move the giraffes into the enclosures using drag and drop. In this way they can visualise the division problem.

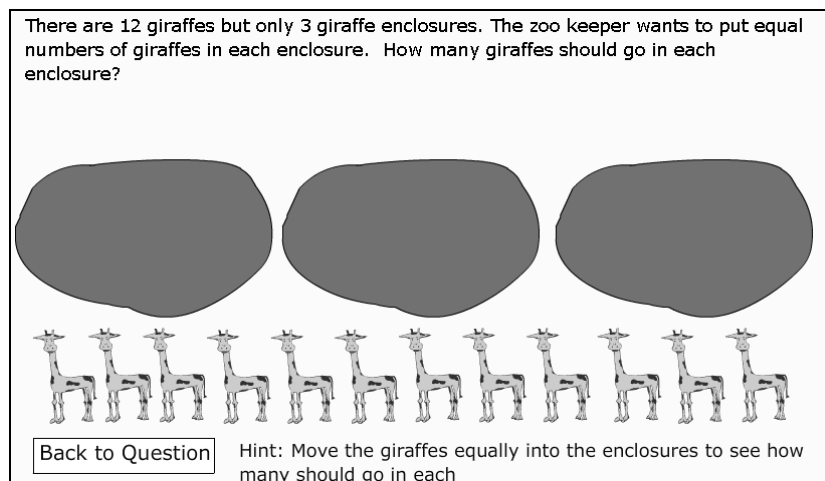


Figure 5.10 Giraffe help activity

Elephant Help Activity

Based on the elephant question (see Table 5.2), the elephant help activity (see Figure 5.11) will feature six elephants and a seemingly limitless number of food buckets, stacked on top of each other. The player will be able to drag buckets over to the elephants until each has three, again allowing them to visualise the multiplication problem.

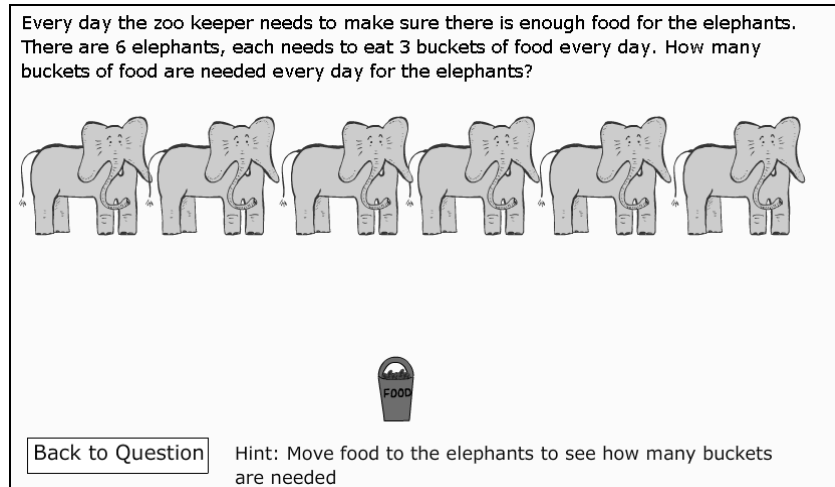


Figure 5.11 Elephant help activity

Lion Help Activity

Based on the lion question (see Table 5.2), the lion help activity (see Figure 5.12) will feature 10 lion huts. The question is a division by ten question where the player has to work out how much each hut should cost. To do this they will be able to enter an amount on each hut, then 'Money Left' will be updated accordingly. In this way, the player should know they have the right answer if there is no money left and each hut costs the same amount of money. In this way the activity allows them to find out the answer through some degree of trial and error.

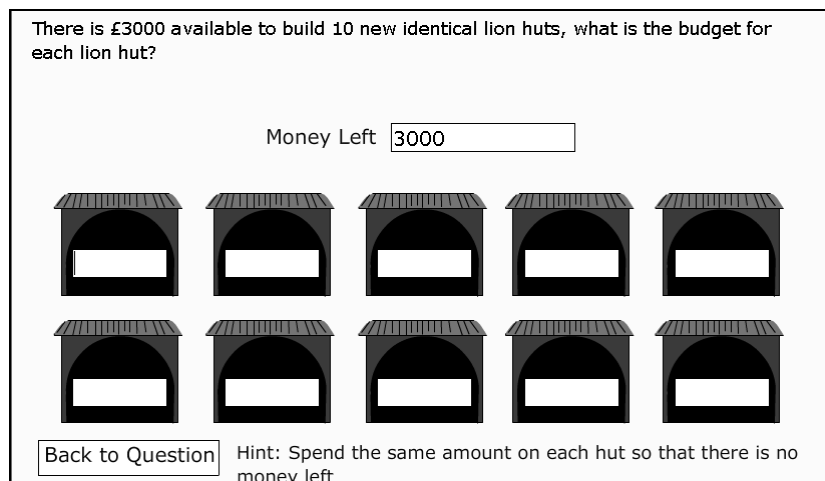


Figure 5.12 Lion help activity

5.4.12 Certificate

The certificate will show the player's name, the date, a rosette and an animal of their choice (see Figure 5.13). For the Super Safari Zone certificate, the player will be able to choose an animal from that zone for the certificate.



Figure 5.13 Maths Zoo Certificate

5.4.13 Printer and Printer Interface

The printer is external to the Maths Zoo system; however, the printing interface is needed so that the player can send their certificate for printing. When the 'Print' button in the menu bar is pressed, the print interface will appear. This will consist of a dialogue box asking "Print certificate?" with 'Yes' and 'No' buttons (see Figure 5.14). Clicking 'Yes' will allow the player to print the certificate and clicking 'No' will just close the dialogue box.



Figure 5.14 Print dialogue box

5.4.14 Open saved game Interface and saved game store

As discussed, the back-end to saving will not be implemented, but the front end will be in place to see how saving would work. When the 'Save' button in the menu is pressed, the save interface will appear (see Figure 5.15), in the form of a dialogue box asking "Save Game?", to which the user can click on a 'Yes' or a 'No' button. 'Yes' will bring up the next dialogue box for them to enter a name and password for the game. Whereas clicking 'No' will just close the dialogue box and return the player to the game.

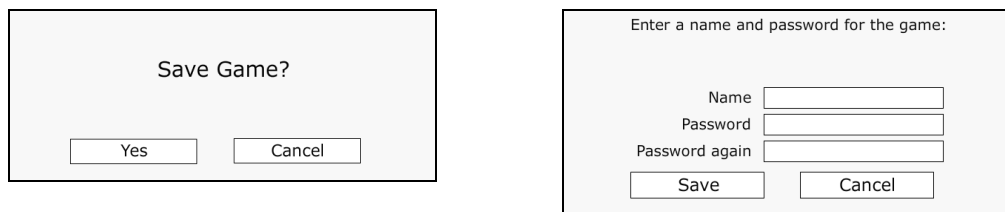


Figure 5.15 Save dialogue boxes

5.4.15 Exit System

When the 'Exit' button in the menu bar is pressed, the exit interface will appear in the form of a dialogue box asking "Exit without saving?" There will be 'Yes', 'No' and 'Cancel' buttons (see Figure 5.16). Clicking 'Yes' will allow the player to exit without saving, 'No' will bring up the save interface where the player enters game details and 'Cancel' will close the dialogue box, returning the player to the game.

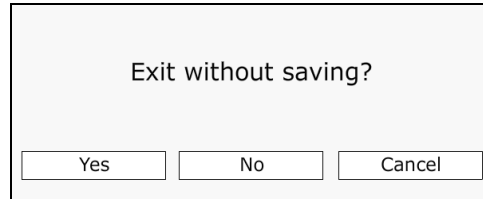


Figure 5.16 Exit dialogue box

5.4.16 On-Screen keyboard

An on-screen keyboard is needed so the game can be used without a hardware keyboard at an interactive whiteboard. The keyboard will have a button for each letter of the alphabet and each number. It will also have delete, caps lock, cancel and return buttons. The letters will be laid out alphabetically because the users may not necessarily be familiar with a QWERTY keyboard, so an alphabetical keyboard is likely to be easier for them to use on the screen.

5.4.17 Colour

A pale yellow will be used as the background (ground) to the system (see Figure 5.17), with the content in much brighter or darker colours (figure), thus giving a high figure-ground contrast. All text will be in black so it stands out from the background. All buttons will have a white background and black text, again to ensure a high figure-ground contrast on the button so that the label can be read easily. Animals and other pictures will be coloured as they are normally and naturally found, so that they are familiar to the user. The zones on the map will each be coloured differently, but using 'natural' colours as would be found outside e.g. greens, yellows, browns and blues.

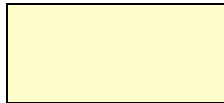


Figure 5.17 Background yellow

5.5 Conclusion

This chapter has detailed high level design decisions that were necessary in order to scope the detailed design of the vertical prototype. Following this it was possible to define the system architecture to get an idea of what different components would be needed in the system and how these should relate to each other. Each component has then been specified at a detailed level such that an initial prototype has been implemented based on this. However, this implemented prototype, known as Maths Zoo Alpha (see Appendix K), does not necessarily meet the needs of the users as they have not been involved since the initial participatory design sessions. Therefore users and stakeholders will be involved again in iterative cycles of evaluation and re-design, such that the final implementation of the prototype is likely to be usable and accepted by the intended users. This will be detailed in the next chapter.

6 User-Centred Re-Designs

6.1 Introduction

Users and stakeholders will be involved in iterative formative evaluations of the high-fidelity prototype. These evaluations are likely to identify usability problems and aspects that could be improved, thus resulting in re-design decisions which will be carried forward into further versions of the prototype. A first phase of evaluations will evaluate the initial prototype (Maths Zoo Alpha, see Appendix K) with children from the design team and a HCI expert. From this, it will be possible to implement a second version of the prototype (Maths Zoo Beta, see Appendix L).

This Beta prototype will then be evaluated with different pairs of year 5 children; year 5 teachers and groups of year 5 children using the prototype on an interactive whiteboard. The design decisions made from this second phase of evaluations will be carried forward to the final implementation of the prototype (Maths Zoo 1.0, see Appendix M). By performing these two phases of evaluations with different users and stakeholders in different formats it should be possible to resolve most usability problems such that the final implementation should be usable and accepted by the intended user group.

6.2 User-Centred Re-Design: Phase One

As mentioned, the first phase of user-centred re-design evaluated the Maths Zoo Alpha prototype with children from the design team and a HCI expert. The processes, findings and resulting re-design decisions from this will now be discussed.

6.2.1 Evaluations with Design Team Children (21-Mar-2006)

Children from the design group were involved in evaluations to assess whether the prototype matched their expectations from the participatory design sessions. The evaluations were carried out in two groups, the first with Children A, C and D; the second with Children B and E. Child F was unavailable at the time of the evaluations. Group evaluations were used because it was thought that the children would be more likely to talk aloud about the system to each other, thus potentially providing more feedback than from an individual evaluation.

At this stage of prototype development, some aspects of functionality and content had not been implemented; this included the questions and mathematical content. This was explained to the children so they would be aware of the game's limitations. However, it was still important to get feedback on these critical aspects of the game. Therefore, the children were instead shown potential questions (see Table 5.2) and associated activities on paper and feedback was taken from this. The children were also asked to carry out a number of high level tasks and to speak aloud about their actions and thoughts, including aspects they liked or disliked. The children were observed carrying out these tasks and questions were asked to obtain additional information.

6.2.2 Evaluations with HCI Expert (22-Mar-2006)

An evaluation was carried out with Sylvie Girard, a student on MSc Human Communication & Computing. Prior to this and following the evaluations with the design team, the questions and activities were implemented. The evaluator was asked to carry out the same tasks as the design group children and to speak aloud about what she was thinking, including any potential usability problems or aspects that could be improved. Questioning was also used where appropriate to elicit further information.

6.2.3 Findings and Re-Design Decisions: Phase One

The findings and re-design resulting from the phase one evaluations will now be discussed in relation to the tasks the evaluators were asked to do on the system:

Task 1: Read the instructions

Child A had difficulty reading aloud several words from the instructions including 'focuses', 'tutorial', 'improve' and 'achievement'. This indicates that some vocabulary used may be too difficult for some children in key stage two, particularly those in lower years and less able readers. Child A also had to use her finger to follow the text and after reading them out said "That was a mouthful", indicating that the instructions were difficult to follow.

The HCI expert also thought some vocabulary used was at too high a level for some children in the intended user group to understand. It was suggested that sentences should be kept direct and short so they are easier to read. Additionally, it was suggested that images should be used where key words and system components are introduced to aid understanding.

Design Decisions from Task 1

- The so-called 'difficult words' will be replaced and re-phrased.
- Sentences will be made shorter with more frequent use of line breaks so the user finds the instructions easier to read and keep their position on the screen
- Pictures of the Map, 'Get Animal Button' from the Safari Zone and the 'Help' button from the menu bar will be included and explained
- The instructions will increase from one to three pages, due to the planned changes
- More instructional text will be added throughout the game in various forms including speech bubbles from the Zoo Keepers and tool tips over some buttons. In this way, the user will be provided with extra information about what to do when playing the game

Task 2: Start a new 'one' player game, for year '5' at 'medium' difficulty level

Some children were confused because of the unavailable 'greyed out' options. This was particularly the case for the 'Number of Players' option as only 'one player' was available and the children worked in pairs or threes. However, once this had been explained to them verbally they appeared to understand and were able to make their selections and continue.

The HCI expert thought these unavailable options may present a usability problem, but that they should remain in the prototype as it is necessary to show what functionality the fully implemented system would have. However, it was suggested that a Zoo Keeper could explain that some of the zoo was 'still being built' and tool tips added to each unavailable option.

The HCI expert thought the 'Cancel' button on the Game Options page should be changed to 'Back' as cancel is not appropriate in that context. She also thought there should be a difference in the appearance between the 'Back' and 'Next' buttons so that 'Next' is a more obvious choice.

On the Choose Zoo Keeper page the HCI expert thought there should be a difference in the colour used to indicate that a Zoo Keeper has been selected and the effect seen when the mouse rolls over a Zoo Keepers. Furthermore, it was proposed that the player should be able to name their selected Zoo Keeper.

Design Decisions from Task 2

- A speech bubble will be added from the Maths Zoo logo saying "We are still building the zoo, so you won't be able to do everything yet, sorry!"
- Tool tips will also be added over each unavailable option to say that the option is not available yet
- The 'Cancel' button will be changed to a 'Back' button; however, the style will not been changed as this would remove some consistency of style from the game
- On the Choose Zoo Keeper page, the roll over effect will be changed to an unfilled square around the zoo keeper. Only when a zoo keeper is selected will the square become persistent and be filled white. In this way, there will be a clear difference between the selected zoo keeper and zoo keepers that have not been selected
- The functionality to name Zoo Keepers will not be implemented because this is unnecessary text input which should be avoided due to Requirement 35

Task 3: Complete a Zone

On the Map, some children commented that there were too many green coloured zones and that other bright colours should be used instead. This is obviously down to their personal preferences, but other children may share this opinion. The colours used are important because they may effect the children's motivation or their interest in playing the game, with brighter colours being more appealing to them.

As scoring had not yet been implemented, the children were asked how this should be done. They thought ten points should be won for every correct answer and five points lost per incorrect answer. The score could then be displayed in the menu bar. They also thought the player could have 'lives', with one life lost for each incorrect answer and the number of lives shown in the menu bar. Also, as a means to display zone progress had not been implemented yet, the children were asked how this should be done. Suggestions were to show the number completed out of the total number of questions or the percentage of questions answered.

When on the Jeep page, the children thought they should be able to choose the jeep colour for the game. The HCI expert evaluation identified that the 'Next' button on this Jeep page was not very obvious and should either be moved to a more prominent screen location or removed, so that the page moves on to the zone without requiring user interaction. However, the children did not seem to have a problem using the 'Next' button to go to the zone.

On the Safari Zone page, the first group of children did not understand the purpose of the 'Get Animal' button until they had read the instructional text on the page. This is a potential usability problem as the actions necessary should be as obvious as possible. The children suggested changing the button label to say 'Click here' or adding an arrow saying 'Click here', so that it is more obvious what to do next.

As the Questions had not been implemented for the design team evaluations this led to some confusion because the questions were not 'real'. However, the children liked the ideas for questions and activities. When asked how the questions could be made more difficult, they suggested using higher numbers and having multiple choice answers closer together. A potential problem was identified by Child D; he thought some children may not understand the word 'enclosure' in the Giraffe Question.

The HCI expert thought the children may not understand the purpose of the 'Help Activity' button, despite it being explained in the instructions. Also, a consistency problem was identified because the button looked out of place in the content area of the screen, so it was suggested that the help activity should be linked to from the menu.

On the lion help activity the HCI expert thought it would take the player a long time to enter the amounts for each of the ten lion huts. Also, the player might enter different amounts when they should be the same. To improve this, it was proposed to have a single text entry box which would populate each lion hut value dynamically. The HCI expert also thought that the zoo keeper could give activity specific instruction in the help activities.

The HCI expert thought the system should give the player a second try if they get a question wrong, but that the system should explain why the answer was wrong. Then if they answer incorrectly again, they should be given a new question.

The HCI expert thought there was usability issue for getting back to the zone when on a question or activity page. At present there is a 'Zone' button in the content part of the page, however, this would probably be better in the menu bar.

Design Decisions from Task 3

- Some green zones will be changed to brighter colours including purple and bright yellow, such that the map will look more aesthetically pleasing to the users
- On the map dynamic text will be added to show the number of questions answered out of total questions in the form of 'Progress: 0/6' over each zone
- The points scheme as proposed by the children will be implemented, with ten points won per correct answer and five points lost per incorrect answer. The score will be shown in the middle of the menu bar so that it can be seen at any point during the game
- The 'lives' feature will not be implemented because deduction of points should be sufficient to discourage the player from getting a question wrong. Also, as the fully implemented game will be quite long, it may be demoralising for the player to despite having answered other questions correctly
- A new page to choose a Jeep from three colours will be added. It will directly follow the choose zoo keeper page as part of starting a new game
- The 'Next' button on the Jeep page will not be changed as the children did not seem to have a problem with this and it gives them the control to choose when to move on

- Arrows will not be used to point to the buttons linking to questions on the Safari Zone page for reasons of consistency as arrows have not been used anywhere else
- The button label linking to the questions on the Safari page will be changed from 'Get Animal' to 'Click here to win animal'
- The 'Help Activity' button from the question page will be removed from the content part of the page. Instead, the Help in the menu will be used because as yet, this has not been utilised. However, this may cause some confusion as the user may associate 'Help' with interaction and game play help not learning help, so this will be explained in the instructions
- The zoo keeper will be added to each help activity with a speech bubble giving instructional text or hints specific to the help activity
- The giraffe question will be changed so that 'enclosure' is replaced by 'field'
- The lion activity will be changed to have a single text input field with the other fields dynamically populated when the 'Enter' button is pressed. A check will also be added to ensure input is a number and to inform the player if it is not, so they can try again
- A link to the 'Zone' will be added to the menu when on the question, thus giving the player more freedom in navigation.

Task 4: Save the game

The children thought they should enter the game name and password at the start of the game, rather than when playing as they may not have time when they need to exit.

Design Decisions from Task 4

- A Game Details page will be added for the player to enter their name, a game name, a password and the password again. Therefore, when the player goes to save the game, they will not need to enter these details
- The front end of the save interface will be changed to reflect this change in process

Task 5: Print a certificate

Some children were confused because the certificate said "Your name here" as the prototype did not yet have the means to get the player's name. Also, some children had difficulty accessing the certificate because they did not notice the 'Get Certificate' button in the content part of the zone page when all animals had been won.

Design Decisions from Task 5

- The certificate will be changed to display the name entered at the start of the game
- A 'Certificate' button will be visible in the menu when a zone has been completed

Task 6: Exit the game

The children did not have any difficulty leaving the game, however the HCI expert thought the process should be changed to a two step exit process. Firstly, the game should ask the player if they want to leave, then ask the player if they want to save.

Design Decisions from Task 6

- There will be a dialogue box to ask "do you want to exit?" and a choice of 'Yes' or 'No' as buttons. If 'No' is chosen, the dialogue box will close and take the player back to the previous point in the game. If 'Yes' is selected it brings up the save game sub-system, however, after saving the game will automatically close

6.3 User-Centred Re-Design: Phase Two

A further iteration of evaluations and re-design were carried out as it was necessary to evaluate the re-designed Maths Zoo Beta prototype (See Appendix L). Also, it was felt that there would be value in evaluating the system with children who were not in the design team and therefore, had never seen the system before. Furthermore, it was thought that evaluations with teachers would be provide useful information as would evaluating the system on an interactive whiteboard with groups of children.

6.3.1 Pair Evaluations (24-Mar-2006)

Three pairs of children from year 5 at Moorlands Junior School were observed and questioned when using the Maths Zoo Beta prototype game for 20 minutes. Each pair consisted of one male and one female participant chosen randomly. The pairs were given tasks to guide their exploration and use of the game. The tasks, findings and design decisions resulting from the pair evaluations are as follows:

Task 1: Read the instructions

All pairs successfully navigated through and read the instructions, however some took longer to do so than others, indicating that some children may be slower or less able readers, however the instructions are now probably written at the right now. Most pairs checked with each other before moving on as to make sure both had finished reading the current page. Most of the children thought the instructions were the right length; however, one child commented that they were too long as children might want to get on with the game.

Design Decisions from Task 1

- The layout of the instruction pages will be re-formatted to condense them from three down to two pages, whilst retaining the same text and graphics.

Task 2: Start a new ‘one player’ game, for ‘year 5’ at ‘medium’ difficulty level

On the options page where the user chooses the number of players, their school year and the difficulty for the game, there seemed to be some confusion. Some pairs were not sure about what they needed to do and had to ask if they should click on each option. One pair seemed particularly confused by the two player option being unavailable despite tool tips and instructional text explaining the unavailability, as added in the previous design iteration. This may have been because they did not notice these things and did not realise that being ‘greyed out’ meant that the option was unavailable.

On the next page where the player must enter their name, a name for their game and a password, there was further confusion. Some children were not sure what they should use as the game name or password. The purpose of this had to be explained to them before they were able to complete this page and move on.

There were no problems on the next two screens where the player has to choose a zoo keeper and a jeep, so all pairs were able to progress quickly to the map.

Design Decisions from Task 2

- The 'number of players' option caused the most problems, so it will be removed from the prototype, however a fully implemented Maths Zoo would include this and possibly allow for other numbers of players
- On the options page, the difficulty label will be changed from 'Difficulty' to 'Game Difficulty', thus making the purpose of the option more explicitly obvious
- A speech bubble will be added to the game details page explaining each field
- The password input will be removed completely as this presented a barrier to starting a new game and a password is not essential because each child has their own network account where the game would be saved

Task 3: Complete a Zone

From the map, most pairs initially went to an 'under construction' zone before being guided to the Super Safari zone by the system, meaning it took them longer to access the questions. This problem is prototype specific because only the Super Safari zone has been implemented. A full implementation of the game would have all zones functioning, so the player would be free to navigate to any zone.

At this stage of prototype development, the Super Safari zone had six animals to win, but only three questions, meaning there was some question duplication. Most pairs did not have any trouble answering the questions, indicating that the questions may not have been challenging enough for them. Pair 1 said the elephant question was easy because it involved the three times table which they were familiar with. To make the questions more difficult they suggested using six, seven, eight and 12 times tables in questions because they are less sure about these.

However, not all pairs found the questions 'easy', Pairs 2 and 3 were unsure of the answer for the elephant question. From doing the help activity, both pairs were able to give the correct answer. Pair 3 also had difficulty with the giraffe question and got the question wrong despite using the help activity. On their second attempt, they used help again and were able to answer correctly.

After winning an animal, there appeared to be some uncertainty for most pairs as to where to navigate next. The menu bar has buttons to go to the zone and the map. Once the pairs realised this, they tended to navigate back to the zone and did not seem to have any further problems with this.

When on the questions, each pair had to be told about the help activity before using it, thus indicating that the pairs may not have known about the help activities or the purpose of them, despite this being covered in the instructions. Alternatively, the help icon may not have been obvious enough for the children to consider using it.

There were some issues related to the usability of the help activities. On the elephant help activity, Pair 1 did not know what to do because they did not realise that the buckets could be moved. They had to be told this verbally, despite the zoo keeper persona giving instructions in a speech bubble. Also, on the lion help activity, one participant in Pair 3 required a verbal explanation of what to do.

Furthermore, on the giraffe help activity, Pair 2 could not get one of the giraffes to 'drop' after it had been 'dragged'. The reason for this cannot be explained as the correct code was in place for this to function correctly. The problem could be related to Flash or the mouse used. Despite this problem, the pair were able to continue the game without 'dropping' the giraffe by using the menu to navigate back to the question.

Another issue related to Flash was the Context Menu coming up with a list of options on click of the right mouse button. This obviously got in the way of game interaction; however, the children were able to recover from it without difficulty.

Design Decisions from Task 3

- For the prototype a 'men at work' road sign icon will be added to each zone that has not been implemented, with a tool tip on mouse roll-over to explain this. This should make it more obvious which zones do not function so that the children do not 'waste time' going to them. This would not be needed in a full implementation of the game
- Although there was some confusion about where to navigate after completing a question, this will not be changed because the children soon learnt through practice what they needed to do. Additionally, having a choice of navigation options gives the player flexibility in the order of game play
- The help activity will be made a more obvious choice by adding a speech bubble from the zoo keeper to remind the player to use the help activity if they are stuck
- The help button will be changed to use colour that makes it stand out more
- Further questions and associated help activities will be developed at a higher level such that the children are provided with an appropriate challenge
- Zebras will be added to the Safari zone so there are more questions and activities to do
- Although there were some issues with some children not knowing what to do initially on some help activities, it has been decided not to change these because the children soon learnt what was necessary. It is also likely that as they become more familiar with the system, this will not be a problem
- The context menu in Flash should be removed if possible, however, when used on interactive whiteboards this context menu should not be problematic as it is difficult to right click unintentionally with an interactive whiteboard pen

Task 4: Save the game

All pairs were able to save the game without trouble. However, the back end to this has not been implemented and will not be for the purpose of the prototype system. In view of this:

- **There are no design decisions from this task**

Task 5: Print a certificate

There were some issues with the pairs navigating to the certificate. When all questions in a zone have been completed, a button linking to the certificate comes up in the menu bar when in that zone. However, none of the pairs noticed this until it was pointed out to them. Pair 3 went directly back to the map when they knew they had won the last animal, so they did not even see this button.

Design Decisions from Task 5

- The greeting in the zone will be changed from a welcome message to a well done message, thus making it more obvious that the zone is complete

- On the map, a button in the form of a rosette icon will appear over the appropriate zone when the certificate has been won. The rosette icon will have a tool tip explaining that the player should click to get their certificate. In this way, the player will be able to see quickly which zones they have won a certificate for and access each certificate directly from the map

Task 6: Exit the game

The pairs were all able to exit the game without any problems. In view of this:

- **There are no design decisions from this task**

From these evaluations, several usability problems have been identified and decisions made for re-design in order to resolve these. However, it has also been realised that the learning content provided by the system may not be at the right level for some children. Therefore it is necessary to evaluate the system with teachers from year 5 so that they can judge this and hopefully makes suggestions to improve this.

6.3.2 Teacher Evaluations

Two year 5 teachers at Moorlands Junior School evaluated the Maths Zoo with the main aim of assessing the system's appropriateness for key stage two children. As the time available for these evaluations was limited, the teachers were shown a demonstration of the system and asked to comment on good aspects and those which could be improved.

Teacher Evaluation (27-Mar-2006)

The feedback was primarily positive; the teacher thought the system was good; that the children would like it and that it would be particularly suitable for lower ability children and children in lower years of key stage two. She said this was due to the highly visual content, which can really help 'visual learners'. Also, she thought that as the written content of the system is quite limited, it would not be a barrier to younger children or less capable readers.

Furthermore, the teacher thought the division by ten questions were particularly suitable for the year 5 group because this is a key learning objectives. However, she thought some other questions may not be challenging enough for high performers and children at the top of key stage two. In view of this, a discussion followed about adapting the question content to provide a suitable challenge for these children, so that they could also benefit and learn from using the system.

One suggestion was to use much higher numbers in the questions. A consequence of this would be that some help activities may no longer be feasible because the visual representations of numbers, in the form of zoo animals and zoo related objects may not fit clearly on the screen. However, it could be argued that these higher ability children would not need the activities because they no longer rely on concrete concepts to solve problem, as a result of them having moved into more abstract levels of thought and problem solving.

A further suggestion was to use numbers that result in a remainder when divided, with the child having to work out the remainder. Another idea was given using an example of egg boxes that fit a certain number of eggs. The child must work out the total number of boxes needed. The teacher thought this can be a particularly good question when the number of

eggs does not divide equally between the boxes, meaning the child must realise an extra box is needed to house the remainder.

Teacher Evaluation (29-Mar-2006)

The feedback from this evaluation was also very positive; the teacher liked the system and thought it would be particularly good for remedial teaching. The teacher thought the system could be used across key stage two, because it is the ability level of a child rather than their school year that dictates what content and material is suitable. Furthermore, the teacher thought Maths Zoo would particularly appeal to girls due to its visual and interactive nature.

A further positive comment about Maths Zoo was that the teacher thought it was clear and easy to navigate; however, she thought children may benefit from using the system with support from an adult on first use. Alternatively, children could work in pairs of different ability levels, with the more able child teaching the other child how to use the system. These ideas link to Vygotsky's *zone of proximal development* theory and Bruner's *scaffolding* concept (see Section 2.3.2)

The teacher also thought that the inclusion of learning objectives was good for teachers. Additionally, she thought the division by ten questions were good because many children have difficulty grasping this concept. For the help activities, the teacher suggested that there should be the ability to drag and drop more than one object at a time because this would help the child's learning, making them think about groups, rather than single units.

Design Decisions from Teacher Evaluations:

- Further questions and associated help activities will be implemented (see Appendix N for a full listing of questions), these extra questions will use higher numbers; get the player to work out remainders; get the player to calculate how many 'boxes' are needed i.e. how many fields are needed to house a certain number of giraffes
- Grouping of objects for drag and drop will be attempted in the implementation

The findings from these teacher evaluations should allow improvements on the learning content to be made. However, the evaluations should not stop here because it is necessary to evaluate the system on interactive whiteboards in view of Requirement 16.

6.3.3 Interactive Whiteboard Evaluations

Two groups of children from year 5 were observed and questioned when using the Maths Zoo game on an interactive whiteboard. The children were given freedom to use the system as they pleased, because it was thought there would be value in observing how the children decided to play the game as a group. One aim of the evaluation was to evaluate how usable the system is on an interactive whiteboard and if there is anything that could be changed to make it more usable. Furthermore, another aim was to see how well the system supports collaboration by a group using it on an interactive whiteboard and if there are any aspects that could be improved that would make collaboration more effective. Following these observations, questionnaires were administered to the participating children to gather qualitative feedback on the system and using it on an interactive whiteboard.

Interactive Whiteboard Evaluation, Group 1: Low Ability (27-Mar-2006)

A group of lower ability children from year 5 used the Maths Zoo on an interactive whiteboard because it was thought that they may benefit from using it the most. The group consisted of six girls, two of which had been involved in the participatory design of the system (Child A and Child C). Child C was very enthusiastic about interacting with the system and took initial control of the whiteboard pen, with the other children taking turns to interact with the game on the interactive whiteboard later.

The instructions were read out aloud to the children so that the group could proceed quickly onto playing the game. Prior to the evaluation, an on-screen keyboard was implemented such that the system could be used without a physical keyboard. The on-screen keyboard was used to enter details for the game. It was noticed that the position of this keyboard on the screen was at just about the right height for the children; however, it may be too high for younger children. This issue is clearly linked to the positioning of the interactive whiteboard on the wall, which is out of the control of a software developer. However, it could be thought that most interactive whiteboards are likely to be located at similar heights in other schools.

A further issue related to the interactive whiteboard was impaired visibility of the content, with colours looking less vibrant, due to light falling on the display. Despite this, the children were still able to read the text and see the objects such that they were able to play the game successfully. However, the visibility did seem to have some effect on the interaction with the game, with some children appearing to be less sure of what to do next in comparison to the trials carried out by pairs at workstations. This is possibly because aspects of the system looked less obvious on the interactive whiteboard. Poor visibility on interactive whiteboards appears to be common problem that is to some extent out of the control of a system developer. However, through the use of colours with high figure-ground contrast, as used with the Maths Zoo, the effects have been reduced.

When on the questions, the group usually read out the questions together; followed by some children verbalising their opinion on the 'right answer'. Quite often, there were differences in opinions meaning some children actually had the 'wrong answer', indicating a lack of understanding about division. When on the first question, the children were reminded to use the help activity if they were not sure of an answer. However, on subsequent questions, the children were quick to go to the help activity and often reminded the person interacting with the game to go to the help activity.

With the giraffe activity, the children moved the giraffes by the drag and drop into different fields. In this way they were able to work out the correct answer by making sure there were equal numbers of giraffes in each field. This activity led to the children discussing the question and how to work out the answer. It was noticed that children who were not directly interacting with the whiteboard frequently gave instructions as to where certain giraffes should be moved. During the activity, some children verbalised a change from their initial 'right answer', indicating that the question visualisation helped them 'see' the correct answer.

These help activities seemed to add value to the children's learning experience, however, there were some issues with the drag and drop on the whiteboard. In order to drag the objects, the child must maintain contact with the whiteboard, only releasing the pen to drop the object. This is clearly more difficult on an interactive whiteboard than with a mouse,

especially as the child may have to move the pen to the other end of the whiteboard before releasing.

Also, with the help activities, it was noticed that sometimes the child interacting would drag an object and place it on top of another one, such that the object underneath could not be seen. This caused some confusion when the children could not find these 'hidden' objects. One child had to go to the workstation to see a more detailed display in order to find them.

The questions the children had the most difficulty with were the 'divide by ten' questions (the lion questions, see Table 5.2). From observing the children, it could be seen that some did not understand how to divide by ten as they were trying all sorts of numbers, some of which were not even close. After several attempts, the children had to be reminded how to divide by 10, then they were able to work out the answer and win the lion.

Interactive Whiteboard Evaluation, Group 2: Mixed Ability (27-Mar-2006)

Another group also used the Maths Zoo prototype on an interactive whiteboard. This time, the children were chosen randomly, rather than based on their ability. The group was mixed gender, with three boys and two girls, all of which were new to the system. This group had similar problems with the interactive whiteboard; however, interaction was more difficult because the pen used did not seem to function perfectly, meaning that the child interacting had to exert more force for it to work.

It could be seen quite quickly that most of the children in the group were quite capable because they were able to work out the answers quickly. For the first question, the children were prompted to use the help; however, this did not really add any value for them as they already knew the answer. The children moved quickly through the questions without making use of the help functionality. This would indicate that the content was not challenging enough for them, which is backed up by some of them saying the game was 'easy'.

Design Decisions from Interactive Whiteboard Evaluations

- The on-screen keyboard will be moved down by about one eighth of the screen to make it more usable to a wider range of users
- The zoo keeper persona will remind the player not to put objects on top of each other so that it is possible to see and count them all
- To design more difficult questions, as already identified in the teacher evaluations

Interactive Whiteboard Evaluation Questionnaire (27-Mar-2006)

Questionnaires were administered to both groups. The number of questions was kept low so that the children would not be put off giving answers. Furthermore, the children were given the option of 'Yes', 'No' or 'Maybe' for each question, thus allowing them to answer the questions easily. The questions and results from the questionnaire can be seen in Table 6.1.

The noticeable results from the questionnaires were that all participants in both groups liked the theme of the game, the look of the game and winning points. The majority of participants also liked winning animals. However, an area where there were some neutral and negative results were around the use of the game on the interactive whiteboard, thus indicating there was some difficulty with this.

Table 6.1 Results from Questionnaire administered after Interactive Whiteboard Evaluations

Question	Group 1 (6 participants)			Group 2 (5 participants)		
	Yes	No	Maybe	Yes	No	Maybe
Did you like the zoo theme?	6	0	0	5	0	0
Did you like the colours and pictures?	6	0	0	5	0	0
Was it easy to use on the interactive whiteboard?	3	1	2	2	0	3
Did you like winning animals?	5	0	1	4	0	1
Did you like winning points?	6	0	0	5	0	0
Did the game help you learn anything?	4	0	2	2	3	0
Did the activities help you work out the answer?	6	0	0	4	1	0

Additionally, there were differences of opinion about whether the game helped with learning. In the first group, two thirds of the participants said ‘yes’ and one third said ‘maybe’. Comparatively, in the second group, the majority said ‘no’ and the remainder said ‘yes’. The difference in opinions for this question between the groups is likely to be linked to the individual abilities within the groups, with the lower ability group seeming to obtain greater learning value from the system. However, the link between ability and the learning value provided by the system will need to be investigated further (see Section 8.3).

6.4 Conclusion

This chapter has detailed the processes, findings and resulting re-design decisions from two phases of user-centred re-design. In the first phase of evaluations, the children in the design team gave feedback on how well the system met their expectations and made design suggestions for aspects not yet implemented. The HCI expert identified usability issues and gave advice on how to handle these. These phases one evaluations have resulted in a series of re-design decisions which have been implemented in a beta version of the prototype (see Appendix L).

The Maths Zoo Beta prototype was then evaluated in phase two evaluations. These involved children new to the system as they were unlikely to be biased. Teacher evaluations provided valuable information around the appropriateness of the question content and advice to make these more suitable. Evaluations with groups of children on interactive whiteboards were used to find out how the system scales up to these and how well it supports collaboration. From this second phase of evaluations, further re-design decisions have emerged. These have been implemented in the final prototype version, Maths Zoo 1.0 (see Appendix M). Details of this implementation will be discussed in the next chapter.

7 Implementation and Testing

7.1 Introduction

This chapter will firstly discuss how the implementation language was chosen, followed by a summary of the main features of Macromedia Flash 8 and Action Script 2.0; the language used for the prototype. The implementation details of Maths Zoo 1.0 will then be discussed.

7.2 Implementation Language Choice

When deciding what language to use for the prototype implementation, it was necessary to identify criteria to assess the options against so that the most suitable one could be chosen. In view of the user group, the requirements and the design, the implementation language should:

- Allow a bespoke Graphical User Interface (GUI) to be developed in order to provide visual learning information (see Requirement 10) as designed
- Allow interaction through direct manipulation (see Requirement 27) with dynamic responses
- Allow for rapid development, in view of the time available for implementation and to allow for several iterations of the prototype with user feedback at each iteration
- Result in software that is accessible in schools (see Requirements 19 and 20)

Based on this, the following implementation language options were considered:

Java Application or Applet

Two types of GUI programs can be developed with Java; standalone applications and applets. An application would be the most appropriate choice because they offer greater functionality and can be used offline. GUI programs often use components from the Swing toolkit, such as buttons and text fields, but there are also custom pre-built components, known as JavaBeans.

Visual programming environment, such as JBuilder, allow screens to be laid out easily thus reducing the programming effort. However, the back-end functionality including listening to and handling events from user interaction must be programmed and there is a high programming overhead to draw complex shapes. It may also be difficult to implement custom components and visual objects with the right look and feel.

Web based program using HTML, PHP and a database

A bespoke GUI could be developed relatively easily as a website if a WYSIWYG development environment, such as Macromedia Dreamweaver, was used. By embedding PHP and linking this to a relational database the content can become dynamic and change based on the user interaction.

However, there is still a significant programming effort for the PHP and database. Also, a website is not necessarily suitable as it would not be possible to access the game without an internet connection. If the connection was slow this may disrupt the flow of play.

Macromedia Flash and Action Script

Macromedia Flash 8 allows different types of GUI and media-rich applications to be developed. It has many features that make it powerful but easy to use. When combined with ActionScript, an object oriented programming language, logic and actions behind the user interface components can be defined. Flash applications can be published and used on a webpage, but this requires the right Flash player to be installed. Alternatively, an application can be packaged as a standalone executable file that includes the appropriate Flash player.

Flash is the most suitable option as it should allow the envisioned design to be implemented relatively easily. Flash allows complex, custom shapes to be developed by drawing directly in Flash or importing external files. By using Action Script, the interface can be made dynamic to respond to user interactions and interaction can be through direct manipulation. Also, Flash applications can be used without an internet connection. In view of these reasons, Flash with Action Script has been chosen to implement the Maths Zoo prototype.

7.3 About Flash

Flash applications are developed in Flash documents (.fla files). Flash documents usually consist of the following parts (see Macromedia 2006a for more information):

7.3.1 The Stage

The stage is the user interface area that is seen on playback. Flash's drawing tools can be used to draw directly on the stage or components can be added to the stage from the library (see Section 7.3.3). Components and drawings can be laid out on the screen either by drag and drop direct manipulation or by specifying X and Y coordinates relative to the upper-left corner of the stage. The size of drawings and components can also be changed by direct manipulation or by specifying width and height measures.

7.3.2 The Timeline

The timeline is used to organise the content and the temporal ordering of the stage. Flash documents can play over a length of time, measured using frames. Specific frames can be labelled and ActionScript can be used to move play to a frame label. An important type of frame is a keyframe; these are used where there are changes to the stage or actions.

The timeline also consists of layers, these are like transparencies that stack on top of one another, each containing different content. The order of layers is important, as those positioned higher up will appear on the stage over lower layers. Layers are useful because they allow content to be accessed across frames and keyframes.

7.3.3 The Library and Symbols

The library is a collection of reusable media elements that can be used on the stage. These elements are known as symbols, each has their own timeline and stage. The symbols in the library can either be created in Flash; taken from Flash's common libraries or imported from an external location. The three main types of symbols are:

- **Graphics** - Static images
- **Movie Clips** - Autonomous objects that can be animated and have their own ActionScript

- **Buttons** - Objects that can change their appearance in response to user interaction. Buttons have four frames, the 'Up' frame represents the button when the pointer is not over it, the 'Over' frame represents the button when the pointer is over it and the 'Down' frame represents the button when it is clicked. The 'Hit' frame is an invisible area on the stage that defines the area on and around the button that will respond to user interaction

In order to use a symbol on the stage and apply actions to it, an instance of the symbol must be created. Instances reference the original symbol; therefore any changes made to the symbol in the library will be seen across all instances.

7.3.4 ActionScript code

ActionScript can be added directly to object instances or to keyframes. However, attaching code directly to an object can make coding and debugging difficult, therefore, it is better practice to put ActionScript code in the same place, usually in an 'Actions' layer.

ActionScript code can also be contained in external files (.as files), often in the form of classes that are kept in packages. This provides some modularity and allows for re-use. To use classes they must be imported by specifying the full package name. They will then be automatically compiled into the published version of the application.

7.4 Prototype Implementation

Specific details of the Maths Zoo 1.0 implementation will now be discussed.

7.4.1 Packages and Classes

A package named 'zoo' was created to organize the classes used in the prototype. The classes are 'Game', 'Zone', 'Animal' and 'Question'. Each encapsulates data about different aspects of a Maths Zoo game, with a one-to-many relationships between the classes (see Figure 7.1).

Question instances store specific question details. These question instances are in turn stored within specific animals instances and the animal instances are encapsulated in specific zone instances. Finally, each of these zone instance are then stored within a game instance. In this way, the game instance can store all necessary information for one game, which could be used to save a game.

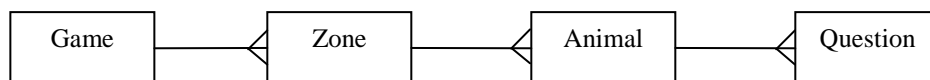


Figure 7.1

Game Class

An object of the Game class (see Figure 7.2) can store specific information about a game. This information is entered and selected when a new game is started and includes the player's school year and game difficulty. Furthermore, a game object can store the player's name, a game name and the player's selected zoo keeper and jeep.

Additionally, a Game object is used to keep track of the game score. This score is initially set to zero then increased by ten points per correct question answer and reduced by five points per incorrect question answer. Moreover, a Game object can store an array of Zone objects, thus allowing further information about a game to be stored and accessed.

```

class zoo.Game {

    //Class properties
    public var _schoolYear:Number;
    public var _difficulty:String;
    public var _gameName:String;
    public var _playerName:String;
    public var _zooKeeper:Number;
    public var _jeep:String;
    public var _gamePoints:Number;
    public var _zoneStore:Array;

    // class constructor function
    public function Game(year:Number, difficulty:String, oneArray:Array){

        this._schoolYear = year;
        difficulty = difficulty;
        this._gameName = "";
        this._playerName = "";
        this._zooKeeper = 0;
        this._jeep = "";
        this._gamePoints = 0;
        this._zoneStore = zoneArray;

    }

}

```

Figure 7.2 Game Class

Zone Class

An object of the Zone class (see Figure 7.3) can store specific information about a zone, including how many animals the zone has and how many have been won. It also stores a score which is used on the zone's certificate. As each zone should focus on a specific mathematical topic, the zone objects stores the zone's objectives, which are then displayed on the Jeep page. A Zone object can also store an array of Animal objects that are in the zone.

```

class zoo.Zone {

    //Class properties
    public var _zoneName:String;
    public var _animalStore:Array;
    public var _noAnimals:Number;
    public var _noAnimalsWon:Number;
    public var _points:Number;
    public var _topic:String;
    public var _objectives:String;

    // class constructor function
    public function Zone(zoneName:String, animalArray:Array,
                        topic:String, objectives:String){
        this._zoneName = zoneName;
        this._animalStore = animalArray;
        this._noAnimals = animalArray.length;
        this._noAnimalsWon = 0;
        this._points = 0;
        this._topic = topic;
        this._objectives = objectives;

    }

}

```

Figure 7.3 Zone class

Animal Class

An object of the Animal class (see Figure 7.4) can store information about specific animals. This includes their type (i.e. elephant or giraffe), gender and whether they have been won yet. An Animal object also stores an array of Question objects that are associated with the animal. The attribute `_questionNo` indicates which question should be used for the animal, it is initially set using the `setRandomStartQuestion`. The value is then changed if the question is answered incorrectly twice, either by incrementing to the next question in the array or wrapping around to the start of the array if already on the last question in the array.

```
class zoo.Animal {  
  
    //Class properties  
    public var _animal:String;  
    public var _sex:String;  
    public var _won:Boolean;  
    public var _questionStore:Array;  
    public var _questionNo:Number;  
  
    // class constructor function  
    public function Animal(animal:String, sex:String, won:Boolean,  
                           questionArray:Array) {  
        this._animal = animal;  
        this._sex = sex;  
        this._won = won;  
        this._questionStore = questionArray;  
        this._questionNo = 0;  
    }  
}
```

Figure 7.4 Animal class

Question Class

Objects of the Question class (see Figure 7.5) store specific questions. Each question needs a value for school year and game difficulty, so that the questions used are tailored to the game options. However, as the system is a vertical prototype restricted to year 5 and medium difficulty, these fields are not utilised, but would be needed in a fully implemented version.

Questions are formed of text and two values that are used in this text. These values are often used to dynamically create the appropriate number of movie clips in a question's help activity. Each question has four multiple choice answers that are stored in an array. These are Strings because answers are not always numeric. The correct answer is also stored which is used to check if player has got the answer right. Each question is limited to two 'tries' and the `_noTries` attribute keeps track of the number of tries a player has had for that question. The question also stores which help activity is associated with it in the `_activityPage` attribute and specific help text for the question and activity.

```
class zoo.Question {  
  
    //Class properties  
    public var _questionText:String;  
    public var _value1:Number;  
    public var _value2:Number;  
    public var _answers:Array;  
    public var _correctAnswer:String;  
    public var _year:Number;  
    public var _difficulty:String;  
}
```

```

public var _helpText:String;
public var _activityPage:String;
public var _noTries:Number;

// class constructor function
public function Question(q:String, V1:Number, V2:Number,
a:String, b:String, c:String, d:String, correct:String,
year:Number, difficulty:String, help:String,
activity:String) {

    this._questionText = q;
    this._value1 = V1;
    this._value2 = V2;
    this._answers = new Array(a, b, c, d);
    this._correctAnswer = correct;
    this._year = year;
    this._difficulty = difficulty;
    this._helpText = help;
    this._activityPage = activity;
    this._noTries = 0;

}
}

```

Figure 7.5 Question class

Importing Classes

It is necessary to import these classes into the Flash document in order to use them. This is done by importing the zoo package:

```
import zoo.*;
```

7.4.2 The Stage

The stage size is specified as 800 by 600 pixels because this is the resolution used on PCs in Moorlands Junior School. The colour of the stage is set to a pale yellow (see Figure 5.17).

7.4.3 The Library and Symbols

The symbols used in the prototype are organised into folders in the library. Most symbols originally started as hand drawn pictures, which were scanned and imported into Flash as bitmap images. These were converted into vector images using Flash's Trace function. They were then coloured and modified using the tools in Flash. After making these changes, these vector images were converted into Movie Clip or Button symbols. By creating instances of these symbols on the Stage, actions could be applied to them.

7.4.2 Layers

The timeline in the prototype uses has the following layers:

- **Labels** - Consists of frame labels for good practice and to aid the development process
- **Actions** - ActionScript layer, keeps the code together for good practice and to aid the development process
- **Menu** - Contains the menu which spans the whole document. The visibility and actions of the menu are controlled using ActionScript in the action later at specific keyframes
- **Jeep** - Jeep animation; the jeep moves across the screen and stops at a road item
- **Content** - Contains most of the content that is displayed on the stage
- **GameData** - ActionScript to generate Question, Animal, Zone and Game objects

7.4.3 Frames, Keyframes and Frame Labels

This frame rate of the timeline is set to 12 frames per second as this is said to give the best results (Macromedia 2006b).

Labelled keyframes are used when there is a change to the stage or ActionScript. Labelled keyframes can then be considered as the 'pages' of the game. Using ActionScript it is possible to move between pages using Flash's gotoAndStop() and gotoAndPlay() functions.

The pages used in the prototype are as follows:

- **Splash** - Opening page, can start a new game, open a saved game or read instructions
- **NewGame** - The player selects the school year and difficulty for the new game
- **PlayerDetails** - The player enters their name and a name for the new game
- **ChooseKeeper** - The player selects the Zoo Keeper for the new game
- **ChooseJeep** - The player selects the Jeep for the new game
- **SavedGame** - Non-functioning in the prototype, but would open a saved same
- **Map** - Shows and links to all zones in the Zoo
- **Jeep** - Animated jeep, also shows zone topics and objectives
- **SafariZone** - Shows animals that have been won or links to questions to win animals
- **OtherZones** - Inform that the selected zone is under construction
- **Question** - Question with multiple choice answer buttons, dynamically populated based on the animal the player is trying to win
- **GiraffeActivity** - Help activity for giraffe questions
- **ZebraActivity** - Help activity for zebra questions
- **ElephantActivity** - Help activity for elephant questions
- **LionActivity** - Help activity for male lion questions
- **LionActivity2** - Help activity for female lion questions
- **QuestionRight** - Informs answer was correct and which animal has been won
- **QuestionWrong** - Informs answer was incorrect and why. Informs the player that they can try again if they still have tries left, else links to a new question
- **Certificate** - Shows certificate with player and game information, allows the player to choose an animal for the certificate
- **Instructions** - First page of the game instructions
- **Instructions2** - Second page of the game instructions

7.4.4 Functionality

Functions and important functionality will now be detailed:

Menu visibility

The menu does not need to be visible on every page of the game. The hideMenu function (see Figure 7.7) hides the menu by setting the _visible property of the menu instance to be false. Conversely, the showMenu function (see Figure 7.8) will make the menu visible by setting the _visible to be true.

```
//Hide menu function
function hideMenu():Void {
    menu._visible = false;
}
```

Figure 7.7 Hide menu function

```
//Show menu function
function showMenu():Void {
    menu._visible = true;
}
```

Figure 7.8 Show menu function

Drag and Drop

The help activities make use of drag and drop functionality. Flash has `startDrag` and `stopDrag` functions to do this, however they must be called separately for each movie clip instance. To simplify this, a `dragAndDrop` function (see Figure 7.9) has been written so that only one function needs to be called. The function starts drag when the mouse is pressed on the instance and stops drag when the mouse is released from the instance.

```
//Drag and drop function
function dragAndDrop(clip) {
    clip.onPress = function() {
        startDrag(this);
    };
    clip.onRelease = function() {
        stopDrag();
    };
}
```

Figure 7.9 Drag and Drop function

On-screen Keyboard

A keyboard movie clip has been developed, with buttons for each letter in the alphabet, the numbers zero to nine, space, delete, caps lock, enter and cancel. The `screenKeyboardEntry` function control they keyboard and aspects of this will be detailed. If `screenKeyboardEntry` is called, it attaches an instance of the keyboard movie clip on a specified layer at an appropriate position on the stage using Flash's `attachMovie` function (see Figure 7.10).

```
//Attach and position the keyboard on layer 105
_root.attachMovie("OnscreenKeyboard", "keyboard", 105);
keyboard._x = 95; //Pixels
keyboard._y = 150;
```

Figure 7.10 Attaching the on-screen keyboard to the stage

By default, 'caps lock' will be off. Pressing the 'caps lock' button will switch caps lock on and off on alternately (see Figure 7.11). The value of caps lock is displayed on the keyboard.

```
//by default have caps lock off
capitals = false;
_root.capitalsVal = "Capitals off";

//Turn caps lock on and off
keyboard.Capitals_btn.onRelease = function() {
    if (capitals == false) {
        capitals = true;
        _root.capitalsVal = "Capitals on";
    } else {
        capitals = false;
        _root.capitalsVal = "Capitals off";
    }
};
```

Figure 7.11 Controlling the caps lock

Pressing a letter or number button on this keyboard causes the associated character to be added to the 'text entered' field, taking into account the caps lock value (See Figure 7.12).

```
//Function to get letter from key press on onscreen keyboard
function keyboardValue(letter, normalLetter:String, capitalLetter:String) {
    letter.onRelease = function() {
        if (capitals == false) {
            _root.textEntered += normalLetter;
        } else {
            _root.textEntered += capitalLetter;
        }
    };
}

//Keyboard values for each letter of the alphabet
keyboardValue(keyboard.A, "a", "A");
//Continues for all other letters...

//Keyboard value for each number
keyboardValue(keyboard.no1, "1", "1");
//Repeated for all other numbers...
```

Figure 7.12 Getting the character value from a button on the keyboard

Pressing the space bar button will add a space after the last character entered. Pressing delete button will remove the last character entered using substrings (see Figure 7.13).

```
//Delete the last character entered
keyboard.Delete_btn.onRelease = function() {
    if (_root.textEntered == undefined) {
        //No text has been entered yet, so don't need to do anything
    } else {
        //Need to remove the last character entered
        _root.textEntered = keyboard.textEntered.text.substr(0,
            keyboard.textEntered.text.length-1);
    }
};
```

Figure 7.13 Deleting characters with the on-screen keyboard

Pressing the cancel button will result in the keyboard movie clip being removed from the screen using Flash's removeMovieClip function, retuning the player to the game whilst disregarding any entered text. Pressing the return button will take the 'text entered' and set this to be the value of the input text field.

Saving Games

The back-end functionality for saving will not be implemented, however, the interface will. The save function will attach a movie clip for the save game dialogue box to the stage using Flash's attachMovie function. If the 'Yes' button on this instance is pressed, the save game dialogue box is removed using Flash's removeMovieClip function and a confirmation dialogue box movie clip attached. This has an 'OK' button, which when pressed will either remove the clip or exit the application if the player had been intending to exit.

Exit Maths Zoo

The exit function will attach and position an exit dialogue box movie clip. On this, clicking the 'Yes' button will confirm the exit and the application will be closed, by calling:

```
fscommand("Quit", "");
```


If the 'No' button is pressed the movie clip is removed and the player returned to the game.

Print the certificate

The printCertificate function will attach and position a print dialogue box movie clip. On this, clicking the 'Yes' button will confirm the print and bring up Flash's print window, from which the printing options can be set and the printing started:

```
print(finalCertificate, "bframe");
```

If the 'No' button is pressed the movie clip is removed and the player returned to the game.

Zone Progress for the Map

The zoneProgress function (see Figure 7.14) gets a zone's topic and calculates its progress in terms of the number of animals won out of the total number of animals. This information is used on the map and is displayed on the map over the appropriate zone button. The function takes a Zone object as the parameter as gets the values for _noAnimalsWon and _noAnimals, which are concatenated into a formatted string. As only the Safari zone has been implemented, all other zones have a progress of '0/0' and do not have a topic.

```
//function to get string with zone progress and topic
function zoneProgress(zoneName):String{
    if (zoneName._zoneName == "Super Safari"){
        progressText = "Topic: " + zoneName._topic + "\n\nProgress:\n"
            + zoneName._noAnimalsWon + "/" + zoneName._noAnimals;
    }
    else{
        progressText = "Progress:\n0/0";
    }
    return progressText;
}
```

Figure 7.14

Zone buttons on the map

Each zone on the map is a button that links to the Jeep page, where the zone is subsequently linked to. The function toZone (see Figure 7.15) associates a zone button to a specific zone and sets the zone to the 'selected zone'.

```
//Function to generate actions for zone button
function toZone(zoneKey, button):Void{
    button.onRelease = function() {
        zoneIndex = zoneKey;
        selectedZone = game._zoneStore[zoneIndex];
        gotoAndPlay("Jeep");
    };
}
```

Figure 7.15

Animal Status in the zone

The animalStatus function (see Figure 7.16) controls which animals and buttons are shown based on what animals have been won and still need to be won in the zone by setting the _visible parameter of each to be true or false accordingly.

```

//Zone status
function animalStatus(animalIndexNo, animalMovieClip, animalButton):Void {
    //Has the animal been won?
    if (selectedZone._animalStore[animalIndexNo]._won == false) {
        //Do not show the animal, show the button to get the animal
        animalMovieClip._visible = false;
        animalButton._visible = true;

        //button to get to question
        animalButton.onRelease = function() {
            selectedAnimal = selectedZone._animalStore[animalIndexNo];
            animalIndex = animalIndexNo;
            gotoAndStop("Question");
        }
    }
    else {
        //Show the animal, do not show the button to get the animal
        animalMovieClip._visible = true;
        animalButton._visible = false;
    }
}

```

Figure 7.16

Check answer

The checkAnswer function (see Figure 7.17) checks if the button pressed is the right answer, if so, the QuestionRight page is opened, otherwise the QuestionWrong page is opened.

```

//Function to check if selected answer is right or wrong
function checkAnswer(answer, answerButton):Void{
    answerButton.onRelease = function() {
        if (answer == rightAnswer) {
            gotoAndStop("QuestionRight");
        }
        else {
            gotoAndStop("QuestionWrong");
        }
    }
}

```

Figure 7.17

Set random start question

The intention is that each animal will have several questions associated with it that are chosen at random, thus giving some variability in game play. The setRandomStartQuestion function chooses a random start question for an animal by generating a random number between zero and the total number of questions for that animal:

```

animal._questionNo = Math.floor(Math.random()*animal._questionStore.length);

```

Limit Flash's context menu

Flash's context menu is used to access and set options associated with Flash, it is usually accessed from an application by right clicking with the mouse. This cannot be removed completely but the number of options can be reduced:

```

Stage.showMenu = false;

```

7.4.5 Publication

Maths Zoo will be published as standalone application (.exe) with its own Flash Player. This will be put onto a CD for use within the school. An Autorun.ini file (see Figure 7.18) will be included on the CD so that the game will open automatically.

```
[autorun]
open=mathzoo.exe
```

Figure 7.18

7.5 Testing

For this sort of system, usability testing has the highest priority and has already been carried out through the iterative evaluations with different stakeholders (see Chapter 6). However, it is also necessary to ensure the system functions as expected and without error. This has been achieved through informal testing throughout the development process, but also with formal black box testing. For this, test cases have been written to define scenarios with specific inputs against expected output. Test cases are passed if the actual results match the expected result. A sample of these test cases and the results can be found in Appendix O.

7.6 Conclusion

The chapter has presented an overview of the implementation of the Maths Zoo 1.0 prototype and has discussed pertinent aspects. However, it has not been possible to detail every aspect and function due to the size of the game, but a full listing of the source code can be found in Appendix V.

Macromedia Flash with ActionScript has been an ideal implementation language due to its highly visual nature and because it allows for direct manipulation, which was essential for the system. Flash has allowed custom graphics and buttons to be developed with ease. However, it has been difficult to follow an object oriented approach because the ActionScript often needs to be in the keyframe where the action occurs. Yet it has been possible to keep most ActionScript in an Actions layer together, which is thought of as good practice.

Also, classes and objects have been used to encapsulate game data. If game saving were to be implemented fully, the Game object could be saved and this would store all information for a game. To extend the game for use in other years and for other difficulty levels, further questions could be designed and then put into Question objects. As Question objects have attributes for year and difficulty level it would then be possible to only use questions that matched the player's selected game options.

Furthermore, to extend the game to cover other topics, the other zones would need to be implemented. As it stands, the prototype already has Zone objects for each zone; however, these would all need animals added to them. These animals in turn would also need to have questions added to them. With these objects in place, the structure of the SuperSafari zone could then be reused and customized for each of the different zones.

Now that the final prototype, Maths Zoo 1.0 (see Appendix M), has been implemented it is necessary to carry out summative evaluations. This will be detailed in the next chapter.

8 Evaluation

8.1 Introduction

Evaluation is an essential aspect of a user-centred design approach. Formative evaluations have been carried out as part of iterative design and re-design cycles (see Chapter 6). Through these evaluations, aspects causing usability problems were identified, re-designed and implemented in the next prototype iteration. This means that most usability problems have been resolved and the final prototype (Maths Zoo 1.0) is likely to be usable for the intended user group. However, it is necessary to carry out summative evaluations to assess the final prototype. These evaluations will attempt to assess the final prototype:

- Against the requirements
- In terms of its learning value for children in year 5
- In terms of user experience

8.2 Requirements Verification

The process of requirements verification involves checking that the system conforms to the requirements specification. Each requirement has been analysed and given a status indicating to what extent the requirement has or has not been met (see Appendix P for the full verification). A 'Met' status indicates that all aspects of the requirement have been met; 'Partial' means some aspects have been met and 'Unmet' means no aspects have been met.

There are 39 requirements; of these 30 have been given a status of 'Met' and nine a status of 'Partial'. There are no requirements that have been 'Unmet'. This is good considering the system is only a prototype. However, it is important to understand why some requirements have only been partially met, particularly those that are considered to have a high priority i.e. those that were specified using 'shall' rather than 'should'.

Requirement 1 relates to providing instruction linked to the national curriculum. This has only been partially met because the prototype only covers division; however, this could easily be extended. Requirements 21 and 22 relate to providing content for each year and ability level in key stage two; this has only been partially met because of high-level design decisions made for the purpose of a prototype (see Section 5.2). Again, it would be possible to extend this. Another high priority requirement that has been partially met is 31. This relates to using high-contrast colour combinations. Although this has been done in most places, there may not be a high enough figure-ground contrast on the map between some of the zones and the zone labels. However, this has not yet caused any usability issues.

The process of requirements verification has been subjective to some extent, but this relates to the nature of the requirements. Requirement 4 states that 'The system should provide a suitably challenging learning evaluation'. This has been classified as partially met because only one level of content has been implemented. Clearly as there are children of different ability levels in year 5, the system will not necessarily provide a challenge for all of them. However, to make this classification less subjective, learning evaluations will be carried out in the next section.

8.3 Learning Evaluation

It is necessary to assess the value of the Maths Zoo prototype for year 5 children in learning the topic of division, as this is the year group and topic the system focuses on.

8.3.1 Planning using the DECIDE Framework

When planning an evaluation, Preece et al. (2002) advocate using the DECIDE framework as a checklist so all necessary aspects can be considered. This framework will now be used to plan the learning evaluations:

Determine the goals

High-level goals are used to guide an evaluation; for this evaluation they are:

- To investigate whether the Maths Zoo prototype has any effect on children in year 5 learning division
- To investigate the direction and extent of any observed effects
- To investigate if there is a link between any observed effects and the mathematical ability level of the children

Explore the questions

To satisfy the identified goals, certain questions must be answered. For this evaluation the questions are:

1. How do the children perform on the topic of division from before, during and after using the Maths Zoo prototype?
2. Are there any changes in performance from before, during and after using the Maths Zoo prototype and to what extent and direction are these?
3. If there are any performance changes, are these linked to practice effects?

Choose the evaluation paradigm and techniques

Based on the goals and questions, the evaluation paradigm can be chosen, which then determines which techniques can be used. The evaluation will be a field study, as it will be carried out at Moorlands Junior School in the natural settings and real world context of the users. Field studies are not usually strongly controlled and tend to be used to understand the natural behaviour of the users; therefore, this type of evaluation may not necessarily allow the high level evaluation goals to be met. Instead, aspects from usability testing, which uses controlled laboratory conditions and set tasks will be applied.

The usability testing technique that will be used is user performance testing, where users perform typical and well-defined tasks under controlled conditions. According to Preece et al. (2002), user testing is most suitable for testing prototypes and working systems, therefore, it should be suitable for use with the Maths Zoo prototype. As user testing has similarities to scientific experiments aspects of experimental design will be considered when planning the evaluations.

Identify the practical issues

It is necessary to identify any practical issues associated with the evaluation prior to carrying out the evaluation. This includes designing the tasks, considering the selection of users, preparing the test conditions and planning how to run the tests.

Design Typical Tasks

There will be approximately 50 children from two year 5 classes to participate in the evaluations, so it is possible to carry out different types of test with the two groups. There will be 45 minutes available for each group, so the tests should be planned with this in mind. The user tests will be carried out as follows:

- **Class 1:** All participants will carry out an individual written test (5 minutes), followed by using the Maths Zoo prototype in pairs (25 minutes), then another individual written test (5 minutes). Both written tests will cover the topic of division; have the same number of questions and have the same format. However, the questions will need to be different but at the same level of difficulty in order to make fair comparisons of participant performance. (See Appendix Q for the tests).
- **Class 2:** Half the class will individually read a one page division summary sheet and do a written test (15 minutes) followed by using the Maths Zoo prototype in pairs (20 minutes), which will include 5 minutes to get familiar with the system. The other half will first use the Maths Zoo prototype in pairs (20 minutes) which will include 5 minutes to get familiar with the system then will individually read a one page division summary sheet and do a written test (15 minutes), thus providing a counterbalance to overcome order and range effects. In this way, all participants will take part in all conditions, but any possible unfair effects of learning from the first task should be neutralized. Only one paper test will be needed and the questions from the paper tests in class 1 will be re-used.

The written tests will feature word problems as this is how the questions appear in the Maths Zoo. The tests will need to be sufficiently challenging so that a ceiling effect is avoided, but should not be too difficult either. The tests will be carried out individually to fit with the normal testing culture in the school and so that any individual changes can be identified. However, limitations with testing are recognised as the results do not necessarily give a true indication of what a child does or does not understand, therefore, the results should be used with care.

For both groups, the participants will use the Maths Zoo in pairs as this fits with how they usually use computers in the school. The pairs will be asked to complete the following:

- Open the Maths Zoo game
- Read the instructions
- Start a new Year '5', 'Medium' difficulty game
- Try to complete the Super Safari Zone
- Write down their score either when the zone is completed or when the time is up

Select Typical Users

It is necessary to involve appropriate users, therefore, it has been planned to use two 'mixed-sex' year five classes from Moorlands Junior School. This should provide a representative set of the user population, however, there are likely to be individual differences between these children which may have an effect on the evaluation results. These differences may include the participants' mathematical ability and computer literacy.

Mathematical ability is an important factor in the evaluation because one of the evaluation goals is to investigate the link between mathematical ability and performance in the user

tests. Within the classes, the children are grouped by mathematical ability, thus making classification of a participant's ability possible. Conversely, it is not so easy to measure a participant's computer literacy so the children should be allocated randomly on this factor. However, it is possible that mathematical ability and computer literacy are linked. Thus allocating on mathematical ability may mean the children are not necessarily allocated randomly in terms of computer literacy, but this cannot really be avoided.

Class sizes are between 26 and 30. As the school's ICT suite only has 16 computers it is normal for the children to work in pairs when using a computer. In view of this and also because the system is intended for peer use, the children will work in pairs when using the Maths Zoo prototype. This leads to potential issues around how these pairs should be allocated. There are several factors that need to be considered for this, including the participants' mathematical ability, gender and relationships with other participants i.e. whether they know each other or are friends.

Firstly, it must be decided if the children should be paired based on similar abilities, different abilities or randomly in terms of ability. In view of the evaluation goals, it has been decided that mathematical ability should be an independent variable that can be manipulated, thus it has been decided to pair the children based on similar abilities, otherwise improvements of performance may be linked to the scaffolding provided by a higher ability child rather than from use of the system. In terms of gender pairings, it has been decided the children should be paired randomly on this factor because it is not one of the evaluation goals to investigate links between gender and performance. It has also been decided that the children will be paired randomly in terms of their relationship to one another as this is something that cannot easily be controlled or measured.

Group 2 will also use pairs for practical reasons; additionally the division of the two sub-groups must be decided. This should be done so that each sub-group is equally matched in terms of ability level as best as possible.

Prepare the testing conditions

It is necessary to attempt to control the testing environment as much as possible to prevent unwanted distractions. However, as the evaluations will be carried out in a school, the extent of control is to some extent limited. The system will be used in the school's ICT suite and a potential issue here is that the Maths Zoo prototype will not function correctly on the school's computers; therefore this should be checked prior to the evaluations.

Problems may arise on the day if there are technical difficulties with the hardware. Also, the children will have to log on to a computer before they can access the Maths Zoo prototype. It has been seen in previous lesson observation that some children have problems with this, so it may be a barrier to them. Therefore, it is important that there is some contingency time available to deal with such problems.

For the written tests it is important that the children work individually so that their individual performance changes can be measured. As there is limited space in the ICT suite, the children are likely to be seated close together. It would be better if the written tests could be carried out in a classroom where there is more space, but this may not be possible. It should be realised that moving the children between rooms part way through the evaluation may cause additional disturbance, which is not necessarily ideal.

Plan how to run the tests

The testing should be run to a prepared schedule with scripts to guide the evaluation (see Appendix R). A pilot test will be run prior to the evaluations (see Section 8.3.2); so that any problems with the evaluation design can be identified and changed. Also, the pilot may uncover practical issues not yet considered.

Decide how to deal with ethical issues

Evaluations involving people need to consider ethical issues, including participant privacy and confidentiality. To protect the participants, their names will not be disclosed or associated with the data collected. It is good practice to tell the participants the goals of the study and exactly what it will involve through a process of informed consent, 'this means that the person giving consent needs to understand why information needs to be shared, who will see their information, the purpose to which it will be put and the implications of sharing that information' (DfES 2006). Verbal informed consent will be achieved by providing the necessary information in the scripts used to run the evaluations. Consent will also be obtained from the children's teacher and the school's head teacher.

Evaluate, interpret and present the data

The resulting data from the evaluations will be the participants' individual scores from the written tests and the pairs' scores from using the Maths Zoo prototype. As this data is quantitative the minimum, maximum and mean can be calculated for the whole class and each of the ability sets. These values can then be compared and analysed to identify if there are any significant changes in performance for each participant.

8.3.2 Pilot Learning Evaluation

A pilot evaluation, in form of the class 1 test, was carried out with a girl from Year 5 from a school in Kent. The pilot was run outside of school, using a laptop. This is quite different to how the evaluations will actually be carried out. In view of this and as only one child was involved there are clearly limitations to what problems and practical issues could be identified.

However, it was possible to see that 20 - 25 minutes should be enough time to complete the Super Safari zone. It also showed that five minutes may not be long enough to answer all ten questions on the written tests. The child in the pilot only answered seven out of the ten questions and of these only three were correct. However it should be remembered that children of different abilities will take the test and there needs to be sufficient scope for higher ability children so that a ceiling effect is avoided. Also, due to the time available for the evaluations it is not possible to increase the time available for the tests. In view of this, the evaluation plan will not change.

8.3.3 Class 1 Learning Evaluation Results and Analysis

A year 5 class with 23 children from Moorlands Junior School was involved in the Group 1 learning evaluation as planned. The children carried out individual written tests (Test 1), then played Maths Zoo in pairs, followed by further individual written tests (Test 2). Full results are in Appendix S.

Table 8.1 Class 1 Mean, Maximum and Minimum Test and Maths Zoo scores

	Test 1 Score (from 10)				Test 2 Score (from 10)				Test Score Difference				Standardised Maths Zoo Score (from 10)			
	All	L	M	H	All	L	M	H	All	L	M	H	All	L	M	H
Mean	3.0	0.8	2.1	6.8	2.6	1.2	1.9	5.3	-0.4	0.3	-0.2	-1.5	6.3	4.7	5.8	8.9
Max	8	2	4	8	9	3	4	9	3	2	3	1	10	6	10	10
Min	0	0	0	4	0	0	0	3	-5	-1	-3	-5	3	3	3	6

The class average was 3 in the first test compared to 2.6 in the second test; meaning there was an average decline of 0.4 (see Table 8.1). Of the 23 children, this breaks down to eight who improved, nine who declined and six who remained the same. However, through application of the Wilcoxon Signed-Rank Test (see Appendix S) it is shown that these changes are not considered significant, thus no valid conclusions can be made on the learning value of the Maths Zoo prototype.

The reason for this may be because a single 25 minute period using the Maths Zoo prototype would not be enough to make a difference. Clearly this should be investigated further through a longitudinal study, possibly with children using the system over a week with tests at the start and end of the week. This was not possible to do in the school as the children only have ICT lessons once a week.

What the results have shown is that most written tests scores were low; five children scored 0 on at least one test. This may indicate a floor effect with the written tests being too difficult for some children to get any questions right. This is particularly evident in relation to two children who required reading assistance and only scored 0 or 1 on both tests; the word problems were clearly a barrier to them. Therefore, it may have been better to use a combination of word problems and number calculations to make the tests more accessible to less able readers. A further issue with the written tests was that some children thought both tests were the same due to the similarity of their layout and the questions. In view of this, it may have been better to have explicitly informed the children that they would be similar but different question.

The results also indicate that a ceiling effect with the written tests has been avoided as the highest score was 9, thus leaving scope for improvement. Yet it is possible that the Maths Zoo has a ceiling effect as two pairs obtained full marks and the lowest score was 3. In comparison to the written tests, the Maths Zoo class average was at least double the written tests average, thus potentially indicating a mismatch in the level of questions. If this was the reason, the written test questions should have been slightly easier and the Maths Zoo questions slightly harder, so that they were more equally matched.

Another reason why the Maths Zoo scores were higher may have been because the system is more motivating than the paper tests or because it provides more assistance. With the help activities scaffolding is supplied to help the children work out the answers, therefore, they may be more likely to get them right. Furthermore, as the children had much longer to use the system to allow for familiarisation, they were probably able to answer more questions. It would seem that with the written tests, most children ran out of time as the average number of questions attempted was only six for both tests. Therefore, it may have been better to increase the time available for the written tests at the expense of the time using the system.

The results also indicate a relationship between a pair's ability level and their performance with the Maths Zoo. Pairs in the low ability set scored lower and none of them achieved the maximum score, unlike in the other sets. From this it could be implied that the children in the low ability set have more scope for improvement with the Maths Zoo, whereas there may not be enough scope for improvement in the other sets as it may be too 'easy' for them.

A further influence on the results may be that the written tests were carried out individually but the game used in pairs. In this was the pairs could have discussed the question and answers so they may have been more likely to get questions right. Clearly further research is needed to investigate if there are differences in performance between individuals and pairs when using the Maths Zoo as there may be positive effects from peer use of the game. This type of investigation was not carried out in the school because they have a culture of pair computer work, thus individual computer use would go against this as well as being unfeasible given the computer resources available.

For the part of the evaluation using the Maths Zoo it took some pairs much longer to get started because they had to log in and some of the computers are quite slow. The computers also caused problems because some of the mice did not work properly so the children could not use the drag and drop functionality in the help activities, meaning they may not have performed as well as they could have.

8.3.4 Class 2 Learning Evaluation Results and Analysis

A year 5 class with 26 children from Moorlands Junior School was involved in the second learning evaluation as planned. Group A consisted of 12 children who carried out written tests individually then played Maths Zoo game in pairs. This was counterbalanced by Group B, which consisted of 14 other children who played Maths Zoo first in pairs then did the written tests individually. Full results can be found in Appendix T.

Table 8.2 Class 2 summary of written tests scores for Group A and B

	Written test score (from 20)							
Set	<i>All</i>		<i>Low</i>		<i>Middle</i>		<i>High</i>	
Group	<i>A</i>	<i>B</i>	<i>A</i>	<i>B</i>	<i>A</i>	<i>B</i>	<i>A</i>	<i>B</i>
Mean	6.2	5.1	1.5	3	7.8	0.8	9.3	9.3
Max	13	13	3	3	10	2	13	13
Min	0	0	0	3	5	0	6	6

Table 8.3 Class 2 summary of Maths Zoo scores for Group A and B

	Standardised Maths Zoo score (from 20)							
Set	<i>All</i>		<i>Low</i>		<i>Middle</i>		<i>High</i>	
Group	<i>B</i>	<i>A</i>	<i>B</i>	<i>A</i>	<i>B</i>	<i>A</i>	<i>B</i>	<i>A</i>
Mean	15	9.4	12.5	4.4	11.3	6.9	19.2	16.9
Max	20	18.8	20	5	15	10	20	18.8
Min	5	3.8	5	3.8	7.5	3.8	17.5	15

Group A can be considered the baseline for the written tests as they did this first. Compared to Group B, there is an average decline of 1.1 (see Table 8.2), so it would appear that Group A performed better than group B who did the written test second. However, by applying the Robust Rank Order Test (see Appendix T) it is shown that these changes are not considered significant, thus no valid conclusions can be made on the prototype's learning value.

The reasons for this are likely to be similar to those from the other learning evaluation (see Section 8.3.3). However, additionally, it is known that some children did not realise there were questions on both sides of the test, meaning that they only attempted questions on the first side. Five children in Group A only answered one side, compared to ten in Group B; however, it cannot be known if this was because they did not realise there was a second side or because they ran out of time. To reduce the likelihood of this occurring, the instructions regarding the tests could have been made more explicit.

Furthermore, the written tests were carried out in the ICT suite because it was not possible to separate the class between two rooms. This meant the children were seated closer together than would have been liked. From looking at the tests it is seen that two pairs had identical answers, so it is likely that they completed the tests together. Therefore, these scores are unlikely to reflect the individual performance for these children, however, not much could have been done to avoid this.

With the Maths Zoo scores (see Table 8.3), there appears to be a relationship between ability set and performance in Group A, with the scores reflecting the ability level. However in Group B who used the system first, the low ability set outperformed the middle ability set on the system and on the written tests. If it is the system that has caused this difference, it may indicate that the system is more suitable, effective or motivating for the low ability set.

8.4 User Experience Evaluation

According to Preece et al. (2002), user experience concerns what the interaction with the system feels like to the users in subjective terms. This is an important area to investigate in view of Requirement 2, which states ‘The system should provide a suitably enjoyable learning experience’ and Requirement 14, which states ‘The system should give rewards valued by the user’. Clearly both of these are important if the system is to be accepted and liked by the user group, but both can only be measured and rated subjectively.

Questionnaires have been used to investigate user experience. These were administered to the year 5 children in pairs directly after they had used the Maths Zoo prototype in the learning evaluations (see Section 8.3). As the time available was limited, only two questions were asked; the first was ‘What did you like about the Maths Zoo?’ and the second ‘What didn’t you like about the Maths Zoo?’ It was hoped that questions like these would give the children freedom to write about what they thought was important.

The full listing of responses can be found in Appendix U. Of the 24 pairs, 23 responded and there were 23 responses for the ‘like’ question. 11 of these made reference to the learning aspects of the game (i.e. the questions, division, maths etc.), ten mentioned the word ‘fun’ and seven made reference to the theme of the game (i.e. winning animals, zoo keepers etc.). These responses indicate the game is considered a suitably enjoyable learning experience and that the users value the rewards as they seem to like the zoo theme.

Interestingly there were two mentions of the game being ‘easy’ as things the children liked. On further analysis it was seen that these responses came from children in the low and middle ability sets and that they had only scored 20 and 50 out of a possible 80 on the game. Therefore it would appear that the game has made mathematics seem easy for them, even though they are not yet getting every question right.

There were 17 responses for the ‘dislike’ question, however, three of these stated ‘Nothing’ and another stated that ‘It was too much fun’, thus leaving 13 responses. Seven responses, all from the low and middle ability sets related to the learning content being too difficult. This is not necessarily bad because Requirement 4 states that ‘The system should provide a suitably challenging learning experience’. Therefore it is a good that the system has given them a challenge.

Conversely, two responses, both from the high ability set related to the learning content being too easy, with the respondents scoring 70 and 80. This would indicate a need for more challenging content for this ability set. There were also two ‘dislike’ responses concerning the usability of the system; one regarding the instructions being unclear and the other stated that the help was difficult to use. Although it is not known why; it would have been useful if these children could have been interviewed.

8.5 Conclusion

This chapter firstly summarised the findings from verification of the requirements, which showed that most requirements have been met otherwise partially met. Of the requirements that were partially met, many were due to the prototype nature of the system. High-level design decisions were made prior to implementation in order to restrict the scope in view of the time and resources available. However, it should be possible to extend the system to meet these requirements as the underlying framework for them is already in place.

Following this, the chapter detailed the planning and results for two different learning evaluations carried out with year 5 children at Moorlands Junior School. The results have not been significant enough to allow conclusions to be drawn on the learning value of the Maths Zoo prototype; however, the need for longitudinal investigations has been identified. It is also suggested that further evaluations could investigate the effects of carrying out paper tests and using the system individually compared to in pairs. This then extends to investigating the effects of different types of pairings, for example, same ability pairs against different ability pairs.

Furthermore, the evaluations have identified potentially influential factors that could be controlled further if the evaluation were replicated. It is suggested that the written tests should have a combination of number and word problems; that the questions are made slightly easier or the time available increased and that the format of the tests are made more explicit to the children as to avoid them thinking the tests are the same.

In addition to these learning evaluations, questionnaires have been used to gather information about the ‘user experience’. This focused on finding out what the children like and dislike about the Maths Zoo prototype. These results have been difficult to analyse because they are qualitative and subjective, however, it can reasonably be concluded that the users like and enjoy using the system which is implied by the numerous mentions of the word ‘fun’ and references to the theme of the game.

9 Conclusion

The essence of the problem tackled throughout this project centres around the need for additional mathematics support in key stage two. Mathematics is a high educational priority, but many children find it difficult to grasp, which has led to a significant number of children not reaching the target level of achievement by the end of primary school. It is thought that mathematics is difficult to learn due to its abstract nature. It can also be hard to teach, particularly as teachers have large mixed ability classes and cannot always give individual tuition. Furthermore, negative attitudes and problems with confidence towards mathematics can present additional barriers to learning.

With a drive to increase Information Communication Technology (ICT) usage in mathematics, it was proposed that a computer based solution could provide this additional support. However, this raises further issues as primary school children rarely use computers individually. Clearly the problem is complex and careful design is needed for a computer based solution to be successful.

To tackle the problem it was firstly necessary to research relevant literature to provide a theoretical grounding to the project and a better understanding of the problem. It was found that children in key stage two are normally in Piaget's (cited by Lee and Das Gupta 1995) *concrete operational* stage of cognitive development, meaning their problem solving is linked to physical reality. A need to link learning to familiar concepts was also identified. Thus it was suggested a computer based solution should attempt to make mathematics more concrete through use of appropriate and familiar visual representations and metaphors. Another key finding was that through instruction and assistance known as *scaffolding* (Wood, Bruner and Ross 1976, cited by Woolfolk 2004), children can solve more difficult problems within a *zone of proximal development*, as theorised by Vygotsky (cited by Woolfolk 2004). In view of this it was suggested that a computer based solution could provide this instruction and assistance.

The literature review also made apparent that current software designed for children can actually hinder collaboration and peer learning because it is intended for individual use and there can be conflict when sharing input devices (Stewart et al. 1998; Monteith 1998). Additional input devices could be used for simultaneous interaction, but implementing this is overly complex and schools may not have the appropriate hardware, so this option was considered no further. However, interactive whiteboards are commonplace in schools and are suited for use by groups, therefore, the project moved to focus on designing for these. This sort of technology should enable peers to interact around the shared interface, which may be beneficial to learning in view of Vygotsky's (cited by Woolfolk 2004) beliefs that social interactions and shared activities are necessary when learning new concepts.

Clearly there are challenges in designing for this medium because traditional Graphical User Interface (GUI) design is thought to be ineffective. Also, the system must function on a Personal Computer (PC) if it is to function on an interactive whiteboard and there may be times when it is not possible to use the system on an interactive whiteboard. In view of this, it was realised that the system should be usable on both PCs and interactive whiteboards; which then presents additional challenges to design. A further consideration is the need to

design learning content that encourages collaboration, so that the children are more likely to work together and can benefit from peer learning.

The literature in isolation was not enough to design such software; for it to be usable and accepted by the intended user group it was of critical importance to involve users throughout the design process. This began by building a picture of the user group through lesson observations and interviews with children and teachers from key stage two; drawing on methods used in ethnography and contextual inquiry as discussed in the literature survey. From this involvement it was found that in Moorlands Junior School (the school involved throughout the project), interactive whiteboards are used in most lessons, however, ICT is rarely used with mathematics, but they would like this to increase; meaning there was scope to increase usage with the proposed software solution.

A problem identified by one of the teachers was that children often lack confidence with mathematics, so they need encouragement and constructive feedback. From this it was inferred that a software solution should attempt to build confidence. On a related matter, the children interviewed had negative attitudes towards mathematics with claims of it being 'difficult' and 'boring'. Therefore, it was decided that a software solution should try to make maths more interesting and fun, which links to the notion of obtaining 'enjoyment of learning' as discussed in the Primary National Strategy (DfES 2003).

Although stakeholder involvement provided valuable information, the validity of this was to some extent limited as only a narrow range of stakeholders were involved. It would have been better to involve teachers and children from all years in key stage so there was a more representative set of stakeholders.

To gather further information as part of establishing requirements, existing mathematics software was evaluated by a HCI expert. Poorly designed features were identified and specifically excluded from the requirements whilst well designed features included. These evaluations resulted in more usability problems than positive aspects being identified; thus indicating problems with existing educational software and scope for improvement.

After establishing requirements, the user-centred approach continued into design. A pilot brainstorming session was used to trial 'designing with children' and generate initial ideas to guide subsequent sessions. With the pilot session, it was possible to identify which methods and approaches yielded the best interaction and results. It was found that the children had trouble moving away from pre-conceived ideas about what software should do. Therefore it was necessary to move the focus away from the final product down to a lower level with concepts the children were more comfortable with. This was done by inquiring about their interests and favourite things, which resulted in more discussion and creative thought.

Following this, a design team with four children from year 5 in Moorlands Junior School were involved in three one hour sessions carried out in the spirit of participatory design. The first involved directed brainstorming, where the focus was in finding out the children's interests so the system could take these into consideration. Notable areas of interest were sports and animals. Based on previous stakeholder involvement it was known that many children are interested in animals so this theme was pursued. By the end of the session, the idea for a 'Maths Zoo' game has emerged, where animals are won by answering maths questions in order to build a zoo.

The subsequent CARD (Muller 2001) session explored this idea further by attempting to formalise the flow of the game by ordering index cards relating to different aspects of the game. However, the literature around CARD did not really provide enough guidance on the practical application of the method, so it was difficult to plan the session. It was found that the children did not really understand the purpose of the different coloured cards and fields on the cards. Instead only one type of card was used and only high-level task and associated low-level detail was captured. Despite this, the session was still valuable, but it should be noted that most information was obtained from the audio recording rather than the resulting artefacts. It is thought that storyboarding could have been used instead of CARD at this point as the children may have been more receptive to the visual and concrete nature of this.

A variation of PICTIVE (Muller 1995, cited by Preece 2002) was used next, with audio replacing video recording for reasons of consent and ‘easier’ data analysis. Using audio recording did not detract from the value of the session because it is thought that it is more important to know what was said, rather than what was done. Also, the children were comfortable being audio recorded as the device was unobtrusive. By listening to the recording aspects that had been missed or misunderstood in the session could be identified.

The children particularly enjoyed the ‘PICTIVE’ session, possibly because of its practical and concrete nature. This indicates that design that uses tangible objects can be more successful with children. By the end of the session, the Maths Zoo idea had evolved into a low-fidelity prototype, with clear designs for the zoo keepers, map, menu and certificate featured in the game. In the time available, it was not possible to design every feature of the game, so subsequent design decisions were made without the design team. However, one hour was an appropriate length of time for the sessions as it was possible to keep the children’s interest and attention in this period.

These participatory design sessions can be considered successful because they resulted in a promising design idea which was taken forward to implementation. This success may be attributed to the way the sessions were run. A crucial aspect of this was planning based on the high-level goals of the process as a whole and the low-level goals of the specific session. Before each session, based on the findings and progress made previously, a list of prompts in the form of keywords and scenarios were prepared to guide the session. This meant that when one idea had been covered and had started to trail off, a new prompt was used to change direction or reset the focus of the session. This made better use of the limited time available and kept the children actively involved and interested. Having a ‘bank’ of prompts meant that the most appropriate one could be used at the right time based on what had already been discussed and the remaining low-level goals. Without these prompts, it would have been difficult to run the sessions and keep the children interested.

Based on findings from the practical application of participatory design and building on existing theory, the following framework is suggested for effective participatory design, particularly with children:

1. Focus

- *High-Level:* Decide the high-level focus for the participatory design sessions i.e. decide the ultimate goal and ideal outputs from the whole process
- *Low-Level:* Determine the low-level focus for a specific session. Use the high-level focus and known resource and time constraints to determine the intermediate goals and intended outputs for the session

2. Plan

- *Materials*: Use the low-level focus to plan what materials and resources are needed for a session (an audio recording device, flip chart paper, coloured pens and sticky notes are suggested as staple materials for most sessions)
- *Prompts*: Use the low-level focus to plan prompts in the form of keywords and scenarios to guide the session

3. Direct

- *Guide*: Use the planned prompts to guide and direct the session
- *Familiar concepts*: Start prompting with familiar concepts to encourage discussion and creativity, then adapt the direction accordingly

4. Adapt

- *Direction*: During the session adapt the direction and decide which prompts to use based on intermediate findings, as to pursue interesting routes and attempt to meet the low-level goals
- *Focus*: After the session decide if the high-level focus should change based on the findings and define the low-level focus for the next session

5. Record and analyse

- *Early*: Record key findings and analyse the data as soon as possible after the session while the information is still fresh and before the next session, so the focus can be adapted suitably

This framework sets out a methodological approach to carrying out participatory design with children; but clearly other factors can influence the success. The children involved are likely to influence the direction and adaptation that occurs during the sessions. This project could have taken a different course had other children been involved. There were individual differences within the team; some children were more articulate and forthcoming than others. One child was particularly shy and really needed prompting before making contributions, whereas others readily responded to the prompts. It is thought that participatory design partners need to be relatively confident and comfortable in expressing their ideas and thoughts, particularly as not all ideas will be used.

The children in the design team were from the same year but were mixed in terms of ability. In this way, each provided their own perspectives, so hopefully the resulting prototype is appropriate for children of different ability levels. However, there may have been value involving children from different years as there will probably be differences in cognitive development and interest that may not have been designed for. Yet there could be challenges involving younger children as the design process may need to be more concrete through further use of tangible objects. Also, more direction would be needed from the adult running the session as the children may be less articulate and confident.

Using the resulting ideas and artefacts from the participatory design sessions, design continued into more detail. It was then necessary to restrict the scope of the prototype in view of the time available. It was decided the prototype would be built for year 5 children of medium ability and covering division. It was known from previous stakeholder involvement that children often struggle with division. Designing the mathematical content was challenging as it was difficult to know what was appropriate, despite the guidance given by the national curriculum. Teachers should probably have been involved again at this stage as they could have given advice drawing on their teaching knowledge and experience.

In terms of implementation, there was some deliberation over which language to use. It was soon realised that Macromedia Flash with ActionScript would be best suited to realising the design envisioned during the participatory design. The prototype features zoo keeper interface personas that give assistance and feedback, thus providing scaffolding. The animals are meant to be familiar and they form the main focus in questions. Each question has an associated help activity that features a visual representation of the question where the player can directly manipulate the objects by drag and drop in order to help them work out the answer. This makes the question concrete with visual metaphors whilst providing scaffolding as the child learns how to work out the answer.

Users were involved once again as soon as the prototype was in a suitably functioning state. Initial evaluations were carried out on an alpha version prototype with children from the design team and a HCI expert. These evaluations identified problems with some of the language used, consistency and regarding the limited functionality of the prototype. This led to a series of re-design decisions which were carried forward thus resulting in a beta version of the prototype.

A further phase of evaluations were then carried out on this beta version on PCs with pairs of children who had not used the system before; two year 5 teachers and two groups of year 5 children using interactive whiteboards. Both teachers really liked the system but thought it would be most appropriate for children in lower years of key stage two or lower ability children in year 5 due to its mathematical content and highly visual nature. The teachers made suggestions to make the questions more difficult and these ideas were carried forward to the final prototype.

The thoughts of the teachers were also backed up in evaluations involving children. It was observed that the system may not provide enough of a challenge as some children were able to answer the questions with ease, without using the help activities. For these children it can be thought that their thinking has already progressed to more abstract levels of thought, meaning that visual and concrete representations are of limited value as they can already perform operations mentally.

However, with a low-ability group it was seen that they did not know the answers immediately. The children expressed to the group what they thought the 'right answer' was, leading to some differences of opinion. To resolve this, they used the help activity, often with the children not directly interacting at the interactive whiteboard suggesting this. Conversation around the problem often continued as the child interacting at the whiteboard moved animals and objects around on the screen whilst other children gave instructions about where they should be moved.

On more than one occasion, the children changed from their initial 'right answer' based on performing the help activity. This indicates that the help activities have brought problems to a level where the children are better able to understand. Also, the help activities have encouraged collaborative discussion and problem solving, particularly when used on an interactive whiteboard. It would appear that these lower-ability children really benefited from the visual and concrete nature of the system, which is likely to be linked to their stage of cognitive development.

These evaluations highlighted issues when moving from PC to interactive whiteboard. The system makes extensive use of drag and drop in help activities; which is suitable when used on a PC. However, it can be more difficult to drag and drop on an interactive whiteboard due to the distance that needs to be covered with the pen in contact with the board. Other than this, the system has scaled up well by keeping most interaction in the bottom half of the screen and using high figure-ground contrast to aid visibility.

Although the evaluations gave some indication towards the prototype's learning value, more formal testing was needed to assess this further. Learning evaluations were carried out with two year 5 classes at Moorlands Junior School. It was hoped these evaluations would show improvements from before and after using the prototype, particularly for lower-ability children. However, the results are not significant enough for such conclusions to be made. In view of this, the need for longitudinal investigations has been identified.

What the learning evaluations have shown is a general relationship between performance with the Maths Zoo prototype and ability level. Children in the low-ability set generally scored the lowest, whereas children in the high-ability set often achieving the top score. This indicates that the content of the Maths Zoo prototype is better suited to the low-ability set as there is more scope for improvement. Also the system may be more motivating for this set because they used the help activities more often meaning they had to 'work' for their reward. Conversely, the high ability set rarely used the help activities and could achieve the reward with little effort, so they may get bored of the system. These ideas link to the problem of the match as discussed in the literature review

The 'user experience' of the Maths Zoo prototype has also been investigated using questionnaires. The responses suggest children in year 5 think the Maths Zoo prototype provides a fun and enjoyable way to learn maths. Some responses for aspects they did not like concerned a dislike of division or the questions being 'too difficult'. However, it is considered a good thing that some children were given a challenging experience as there is then scope for improvement, which links with Vygotsky's zone of proximal development theory. Also, some children were disappointed that only one zone had been implemented as they wished to continue playing. Other children asked if the game would be available in the shops; which may indicate that they wanted to play again.

It is also thought that the resulting prototype is a great improvement to the current use of ICT for mathematics in Moorlands Junior School. If the Maths Zoo was extended to a full implementation it could well be an appealing and motivating additional learning resource.

9.1 The Future

With such positive user experience responses, it seems natural that further work could be undertaken to extend the Maths Zoo prototype into a fully functional implementation. To do this, it would be necessary to design further content appropriate for each year in key stage two and for each ability level. However, it is known that it is not necessarily the school year a child is in that influences their capabilities. There is normally a crossover of abilities between the years. Therefore, the system could focus on providing content for the different ability levels, perhaps in-line with the levels outlined in the national curriculum. Also, the scope of mathematical content would need extending to cover other topics in the curriculum.

It is likely that the style of question and interaction would need to be suitably adjusted for different levels and topics. With topics in the 'Shape, Space and Measures' area of the national curriculum, there may be real power utilising the visual drag and drop direct manipulation as already used in most help activities.

It was observed in the evaluations that some year 5 children had difficulty reading. This would probably be more common in lower years; so to facilitate this, questions could be simplified or read out through audio output; however this may not be appropriate in a busy classroom environment. Alternatively, the questions could be made more like the help activities so they can be interpreted visually and should be easier to understand and answer.

Challenges are also faced in designing mathematical content for higher ability levels. It is likely that as the questions become more difficult, so too will designing the help activities. But it can also be argued that children in these higher sets may not need these 'concrete' help activities as their thinking and problem solving will have already moved to more abstract levels in line with their cognitive development. To make the questions more challenging, multiple choice answers could be replaced by input fields and the questions themselves would need to become more abstract.

A further area of expansion relates to providing explicit support for multiple players with a collaborative rather than competitive nature. Questions could be developed with composite parts, such that each player is responsible for one part. If multiple input devices were used with the Single Display Groupware model, then interaction could occur simultaneously. Then if the system was used on an interactive whiteboard, the players could complete the help activities together by each moving objects on different areas of the display.

To develop this content further, a continued user-centred approach is recommended. Different design teams for each ability level could be set up to design the content for their level. Teachers could also become an integral part of the design team as they are teaching experts. However, the teachers need to make sure they are not seen as authority figures because this may restrict the creativity of the children.

For these sessions, it is suggested that the framework proposed in the previous section is used. The high-level focus will be on designing additional content and the low-level focus will be specific to the type of content to be designed in the session. Given that the Maths Zoo theme is well established, it is suggested that any further participatory design sessions should focus on detailed design. Therefore, PICTIVE, or its audio variant are highly recommended.

User involvement should then continue with iterative cycles of design, evaluation and re-design to ensure the system is usable and liked. In addition to this, further longitudinal learning evaluations could be carried out to investigate the learning value of the system for the different years, ability levels and mathematical topics. Another interesting area of research could be around the effects of peer against individual learning with the system, including a consideration of different types of pairings. Evaluations could also research motivation and attitudes towards mathematics after using the game. Further to this, the long term appeal of the game could be explored and if the results were positive, there may be potential for the game to be used by children outside of school. In this way their exposure to mathematics would be increased which may lead to a better knowledge and understanding of mathematics.

10 Bibliography

ASKEW, M., WILIAM, D., 1995. *Recent Research in Mathematics Education 5 - 16*. London: Office for Standards in Education (OFSTED)

ATLANTIC SYSTEMS GUILD LIMITED. *Volere Requirements Specification Template* [online] Available from: <http://www.volere.co.uk/template.htm#anchor326763> [Accessed 05 Feb 2006]

BAKER, K., GREENBERG, S., GUTWIN, C., 2002. Empirical Development of a Heuristic Evaluation Methodology for Shared Workspace Groupware. *CSCW '02: Proceedings of the 2002 ACM conference on Computer supported cooperative work*, November 16-20 2002 New Orleans, Louisiana. New York: ACM Press, pp.96- 105.

BRITISH EDUCATION COMMUNICATIONS AND TECHNOLOGY AGENCY (Becta), 2003. *What the research says about interactive whiteboards*. Coventry: Becta, (BEC1-15006).

BRITISH EDUCATION COMMUNICATIONS AND TECHNOLOGY AGENCY (Becta), 2004, *Getting the most from your interactive whiteboard a guide for primary schools*. Coventry: Becta, (BEC1-15090)

CURRICULUM ONLINE, 2006. *What are eLC's?* [online] Available from: <http://www.curriculumonline.gov.uk/AboutELCs/WhatareeLCs.htm> [Accessed 06 May 2006]

DEPARTMENT FOR EDUCATION AND SKILLS (DfES), 2003. *Excellence and enjoyment, a primary strategy for schools*. Nottingham: DfES Publications, (DfES0377/2003).

DEPARTMENT FOR EDUCATION AND SKILLS (DfES), 2006. *Information sharing: Practitioners' guide*. Nottingham: DfES Publications, (0338-2006BKT-EN)

DIX, A., FINLAY, J., ABOWD, G.D., & BEALE, R., 2004. *Human-Computer Interaction*. 3rd ed. Singapore: Pearson Education, Ltd.

DRUIN, A., 1999. Cooperative Inquiry: Developing New Technologies for Children with Children. *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*, 15-20 May 1999 Pittsburgh, PA. New York: ACM Press, pp.592-599.

EDUCATIONCITY LTD, 2006. *EducationCity.com*. [online] Available from: <http://www.educationcity.com/> [Accessed 14 March 2006]

GUTWIN, C., GREENBERG, S., 2000. The Mechanics of Collaboration: Developing Low Cost Usability Evaluation Methods for Shared Workspaces. *Proceedings of the 9th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, 4-16 June 2000. Washington, DC USA: IEEE Computer Society, pp.98-103.

- HALL, I., & HIGGINS, S., 2005. Primary School students' perception of interactive whiteboards. *Journal of Computer Assisted Learning*, 21, pp.102-117.
- HOPKINS, C., GIFFORD, S., PEPPERELL, S., ed., 1999. *Mathematics in the Primary School*. 2nd ed. London: David Fulton Publishers
- LAFRENIÈRE, D., DAYTON, T., MULLER, M.J., 1999. Variations of a theme: card-based techniques for participatory analysis and design. *CHI '99: CHI '99 extended abstracts on Human factors in computing systems*, May 15 - 20 1999 Pittsburgh, Pennsylvania. New York: ACM Press, pp.151-152.
- LEE, V., & DAS GUPTA, P., ed., 1995. *Children's Cognitive and Language Development*. Oxford: Blackwell Publishers.
- MACROMEDIA, 2006a. *Flash 8 Documentation: What is Flash*
Available from: <http://livedocs.macromedia.com/flash/8/main/00000015.html>
[Accessed 04 Apr 2006]
- MACROMEDIA, 2006b. *Flash 8 Documentation: About frame rates*
Available from: <http://livedocs.macromedia.com/flash/8/main/00000619.html>
[Accessed 13 Apr 2006]
- MONTEITH, M., ed., 1998. *IT for Learning Enhancement*. The Netherlands: Swets and Zeitlinger Publishers.
- MULLER, M.J., 2001. Layered Participatory Analysis: New Developments in the CARD Technique. *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, May 15-20 1999 Pittsburgh, Pennsylvania. New York: ACM Press, pp.90-97.
- NATIONAL CURRICULUM ONLINE. *About the National Curriculum: Key stages 1 and 2* [online] Available from: http://www.nc.uk.net/nc_resources/html/ks1and2.shtml
[Accessed 15 Nov 2005a]
- NATIONAL CURRICULUM ONLINE. *Mathematics Key Stage 2* [online]
Available from: <http://www.nc.uk.net/> Navigate to Mathematics, then Key Stage 2
[Accessed 29 Nov 2005b]
- O'DONNELL, A.M., KING A., ed., 1999. *Cognitive Perspectives on Peer Learning*. New Jersey: Lawrence Erlbaum Associates.
- OFSTED, 2004. *2004 Report: ICT in schools – the impact of government initiatives, Primary schools*. London: OFSTED Publications, (HMI 2050)
- OFSTED, 2005. *The national literacy and numeracy strategies and the primary curriculum*. London: OFSTED Publications, (HMI 1973)
- PRIMARY GAMES LTD, 2006. *Primary Games* [online]
Available from: <http://www.primarygames.co.uk/> [Accessed 29 Nov 2005b]

PREECE, J., ROGERS, Y., SHARP, H., BENYON, D., HOLLAND, S., CAREY, T., 1994. *Human-Computer Interaction*. Massachusetts: Addison-Wesley

PREECE, J., ROGERS, Y., SHARP, H., 2002. *Interaction Design*. New York: John Wiley & Sons, Inc.

REKIMOTO, J., 1998. A Multiple Device Approach for Supporting Whiteboard-based Interactions. *CHI '98: Proceedings of the SIGCHI conference on Human factors in computing systems*, April 18 - 23 1998 Los Angeles, California. New York: ACM Press, pp.344-351.

RUDD, J., STERN, K., ISENSEE, S., 1996. Low versus High-Fidelity Prototyping Debate. *Interactions*, 3(1), pp.76-85.

SANTANEN, E.L., BRIGGS, R.O., & DE VREEDE, G.J., 1999. A Cognitive Network Model of Creativity: A Renewed Focus on Brainstorming Methodology. *ICIS '99: Proceeding of the 20th international conference on Information Systems*, December 12-15 1999 Charlotte, North Carolina. Atlanta, GA: Association for Information Systems, pp.489-494.

SCHULER, D., NAMIOKA, A., ed., 1993. *Participatory Design Principles and Practices*. New Jersey: Lawrence Erlbaum Associates, Inc.

SHNEIDERMAN, B., 1998. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. 3rd Ed. Massachusetts: Addison-Wesley.

SIEGEL, S., CASTELLAN Jr, N.J., 1988. *Nonparametric Statistics for the Behavioural Sciences*. 2nd ed. New York: McGraw-Hill

SOMERVELL, J., WAHID, S., MCCRICKLAND, D.S., 2003. Usability Heuristics for Large Screen Information Exhibits. *Human-Computer Interaction INTERACT '03: IFIP TC13 International Conference on Human-Computer Interaction*, September 1-5 2003 Zurich, Switzerland. Amsterdam, The Netherlands: IOS Press, pp.904-907.

STEWART, J., RAYBOURN, E.M., BEDERSON, B.B., DRUIN, A., 1998. When Two Hands Are Better Than One: Enhancing Collaboration Using Single Display Groupware. *CHI '98: CHI 98 conference summary on Human factors in computing systems*, April 18-23 1998 Los Angeles, California. New York: ACM Press, pp.287-288.

STEWART, J., BEDERSON, B.B., & DRUIN, A., 1999. Single Display Groupware: A Model for Co-present Collaboration. *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, May 15-20 1999 Pittsburgh, Pennsylvania. New York: ACM Press, pp.286-293.

TSE, E., GREENBERG, S., 2004. Rapidly Prototyping Single Display Groupware through the SDGToolkit. *CRPIT '04: Proceedings of the fifth conference on Australasian user interface*, 2004 Dunedin, New Zealand. Darlinghurst, Australia: Australian Computer Society, Inc., pp.101-110.

WOOLFOLK, A., 2004. *Educational Psychology*. 9th Ed. Massachusetts: Pearson Education, Inc.

WOOD, D., 1998. *How Children Think and Learn*. 2nd Ed. Oxford: Blackwell Publishers.

11 Appendices

A	Eight Golden Rules of Interface Design	104
B	Usability Heuristics.....	105
C	The Mechanics of Collaboration.....	105
D	Education City Screen Shots.....	106
E	Primary Games Screen Shots.....	109
F	Pilot Brainstorming Session Spider Diagrams	110
G	Brainstorming Session Spider Diagrams	111
H	CARD Session Ordered Cards	112
I	PICTIVE Session Low-Fidelity Prototype.....	113
J	Initial Game Instructions Text	114
K	Maths Zoo Alpha Prototype.....	115
L	Maths Zoo Beta Prototype	122
M	Maths Zoo 1.0 Prototype.....	131
N	Full Listing of Maths Zoo Questions	138
O	Test Case Sample	139
P	Requirements Verification	141
Q	Learning Evaluation Written Division Tests.....	146
R	Learning Evaluation Scripts.....	147
S	Learning Evaluation Results: Class 1	148
T	Learning Evaluation Results: Class 2	150
U	User Experience Questionnaire Results	153
V	Source Code.....	154

A *Eight Golden Rules of Interface Design*

Shneiderman (1998, pp.74-75) presents eight underlying principles of design that are applicable in most interactive systems. These are:

1. **Strive for consistency** - Consistent sequences of actions should be required in similar situations; identical terminology should be used in prompts, menus and help screens; consistent colour, layout, capitalization, fonts and so on should be employed throughout. Exceptions, such as no echoing of passwords or confirmation of the delete command, should be comprehensible and limited in number.
2. **Enable frequent users to use shortcuts** - Abbreviations, special keys, hidden commands, and macro facilities are appreciated by frequent knowledgeable users. Short response times and fast display rates are other attractions for frequent users.
3. **Offer informative feedback** - For every user actions, there should be system feedback. For frequent and minor actions, the response can be modest, whereas for infrequent and major actions, the response should be more substantial. Visual presentation of the objects of interest provides a convenient environment for showing changes explicitly.
4. **Design dialogue to yield closure** - Sequences of actions should be organized into groups with a beginning, middle, and end.
5. **Offer error prevention and simple error handling** - As much as possible, design the system such that users cannot make a serious error. If users make an error, the system should detect the error and offer simple, constructive, and specific instructions for recovery. Erroneous actions should leave the system state unchanged, or the system should give instructions about restoring the state.
6. **Permit easy reversal of actions** - As much as possible, actions should be reversible. The units of reversibility may be a single action, a data-entry task, or a complete group of actions.
7. **Support internal locus of control** - Experienced operators strongly desire the sense that they are in charge of the system and that the system responds to their actions. Surprising system actions, tedious sequences of data entries, inability or difficulty in obtaining necessary information, and inability to produce the action desired all build anxiety and dissatisfaction.
8. **Reduce short-term memory load** - The limitation of human information processing in short-term memory requires that displays be kept simple, multiple page displays be consolidated, window-motion frequency be reduced, and sufficient training time be allotted for codes, mnemonics, and sequences of actions. Where appropriate, online access to command-syntax forms, abbreviations, codes, and other information should be provided.

B Usability Heuristics

Neilsen (2001, cited by Preece et al. 2002 p.27) and his colleagues developed ten main usability principles. These are:

1. **Visibility of system status** - Always keep users informed about what is going on, through providing appropriate feedback within a reasonable time.
2. **Match between system and the real world** - Speak the users' language, using words, phrases and concepts familiar to the user, rather than system-oriented terms.
3. **User control and freedom** - Provide ways of allowing users to easily escape from places they unexpectedly find themselves, by using clearly marked 'emergency exits'.
4. **Consistency and standards** - Avoid making users wonder whether different words, situations, or actions mean the same thing.
5. **Help users recognize, diagnose and recover from errors** - Use plain language to describe the nature of the problem and suggest a way of solving it.
6. **Error prevention** - Where possible prevent errors occurring in the first place.
7. **Recognition rather than recall** - Make objects, actions, and options visible.
8. **Flexibility and efficiency of use** - Provide accelerators that are invisible to novice users, but allow more experienced users to carry out tasks more quickly.
9. **Aesthetic and minimalist design** - Avoid using information that is irrelevant or rarely needed.
10. **Help and documentation** - Provide information that can be easily searched and provides help in a set of concrete steps that can easily be followed.

C The Mechanics of Collaboration

Gutwin and Greenberg (2000) propose that the mechanics of collaboration are 'small-scale actions and interactions that group members must carry out in order to get a shared task done'. They have identified seven mechanics of collaboration, these are:

1. **Explicit communication** - Group members intentionally provide each other with information.
2. **Consequential communication** - Group members pick up information that is unintentionally "given off" by others as they go about their activities.
3. **Coordination of action** - Group members organize their actions in a shared workspace so that they do not conflict with others.
4. **Planning** - Group members divide and re-divide the task as they go along, reserving areas of the workspace for their use, or considering various courses of action by simulating them in the workspace.
5. **Monitoring** - Gathering information about others in the workspace, including awareness information and more explicit monitoring.
6. **Assistance** - Group members provide help to one another when it is needed. Assistance may be opportunistic and informal, or it may be explicitly requested.
7. **Protection** - Preventing other group members from inadvertently altering or destroying another group member's work.

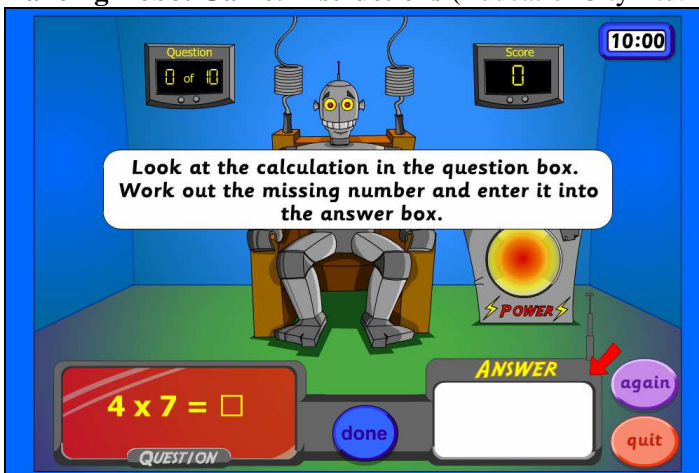
D Education City Screen Shots

Education City Home Page (EducationCity Ltd. 2006)



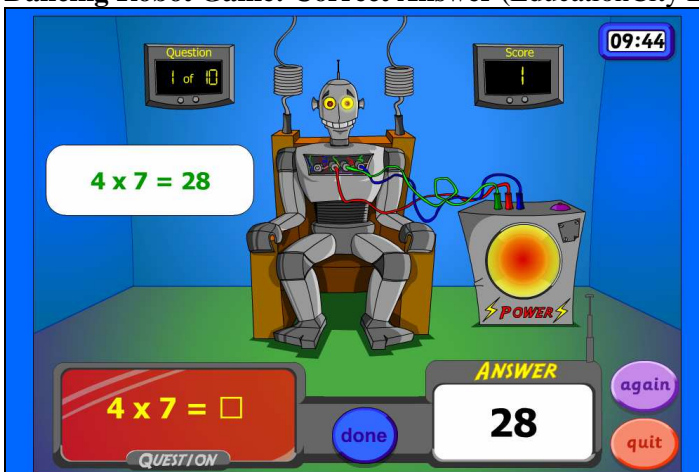
From the home page it is necessary to login before any games can be accessed, however the position and colour of the login button mean that this button is not very obvious.

Dancing Robot Game: Instructions (EducationCity Ltd. 2006)



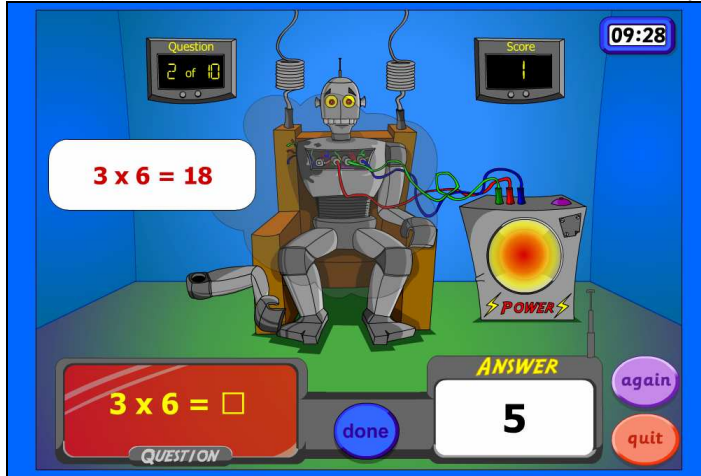
The player is informed of what they need to do and an arrow indicates where the input should go. It would be better if this input could be made directly into the sum.

Dancing Robot Game: Correct Answer (EducationCity Ltd. 2006)



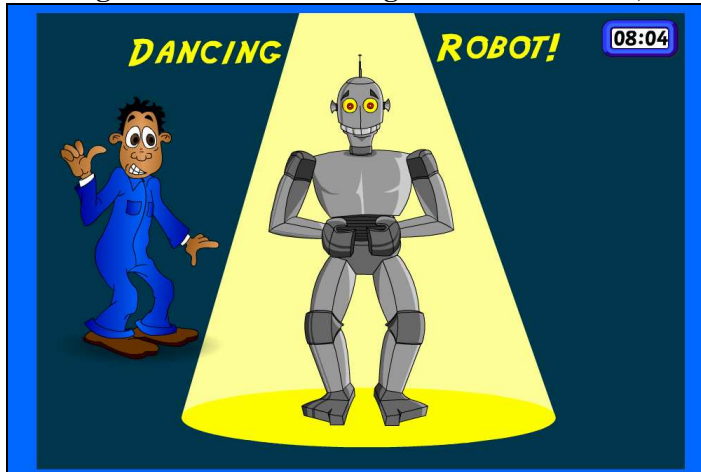
A correct answer is indicated by green text and the robot remains intact. The use of green may not be suitable for a colour blind user.

Dancing Robot Game: Incorrect Answer (EducationCity Ltd. 2006)



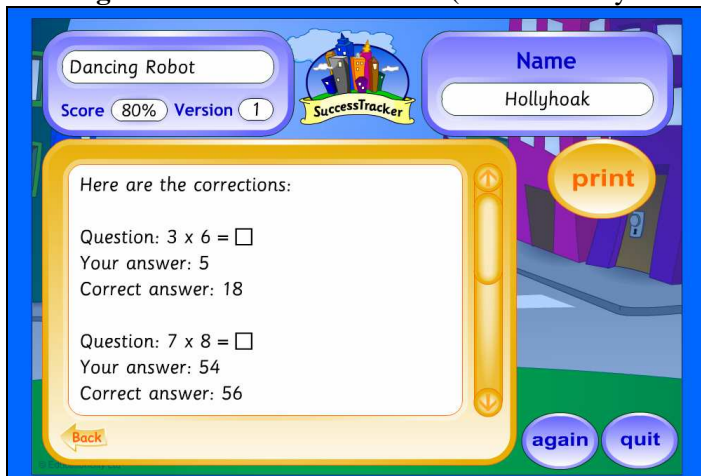
An incorrect answer is indicated by red text and the robot loses a body part. The use of red may not be suitable for a colour blind user.

Dancing Robot Game: Dancing Robot Animation (EducationCity Ltd. 2006)



If the player has scored enough points, the reward on completion of the game is the dancing robot animation; however, it is not shown for long enough.

Dancing Robot Game: Corrections (EducationCity Ltd. 2006)



At the end of the game, corrections can be viewed so the player can see where they went wrong.

Dancing Robot Game: Play again dialogue box (EducationCity Ltd. 2006)



At the end of the game, the system will ask the player 'Again?', but it would be better to ask 'Do you want to play again?'

Slam Dunk Game: Word (EducationCity Ltd. 2006)



The player is shown the word they need to complete by shooting basket balls with letters on them into the relevant hoops.

Slam Dunk Game: Shoot (EducationCity Ltd. 2006)

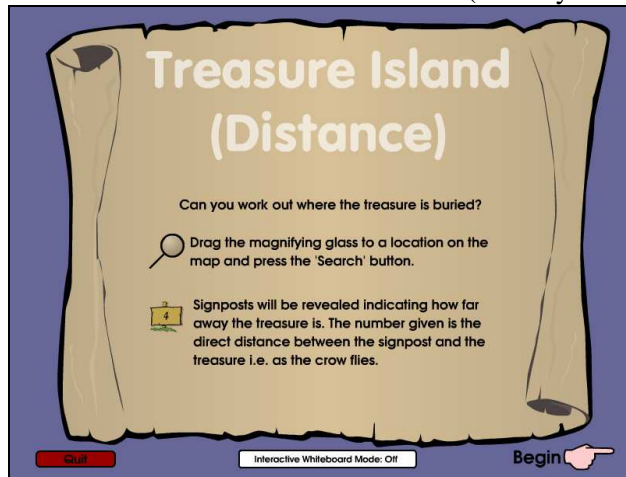


The player is given one ball at time and then has to move between baskets before shooting. Correct shots are indicated by the letter appearing on the basket.

The speaker icon in the bottom left corner will repeat the word.

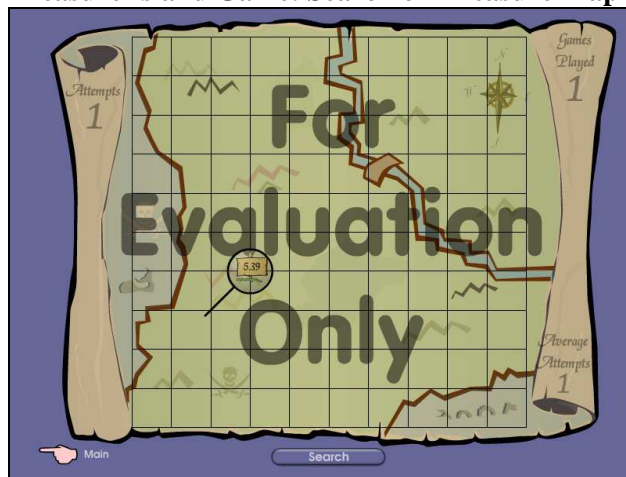
E Primary Games Screen Shots

Treasure Island Game: Instructions (Primary Games Ltd. 2006)



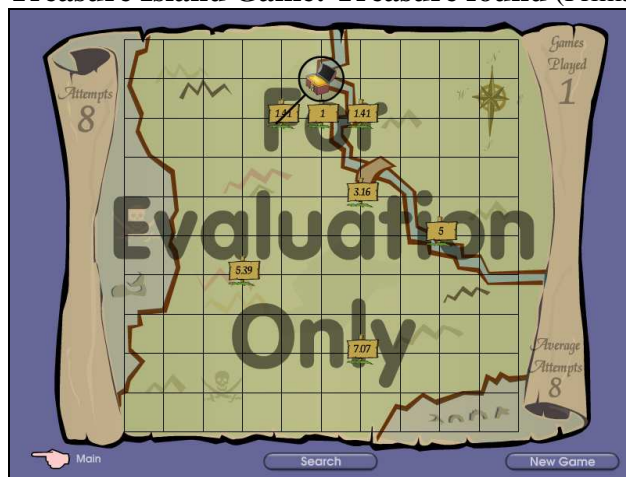
Before playing the game, the player is shown the instructions. The HCI expert thought the instructions may have been at an inappropriate level for younger children, so there should also be an audio version.

Treasure Island Game: Search on Treasure Map (Primary Games Ltd. 2006)



The player has to drag the magnifying glass to a position on the map, then press search. This then shows a sign post indicating how far away the treasure is.

Treasure Island Game: Treasure found (Primary Games Ltd. 2006)



When the treasure has been located, a treasure box will appear on the map to indicate this. It can also be seen how many attempts were needed to find the treasure, but this text does not use a high enough figure-ground contrast.

F Pilot Brainstorming Session Spider Diagrams

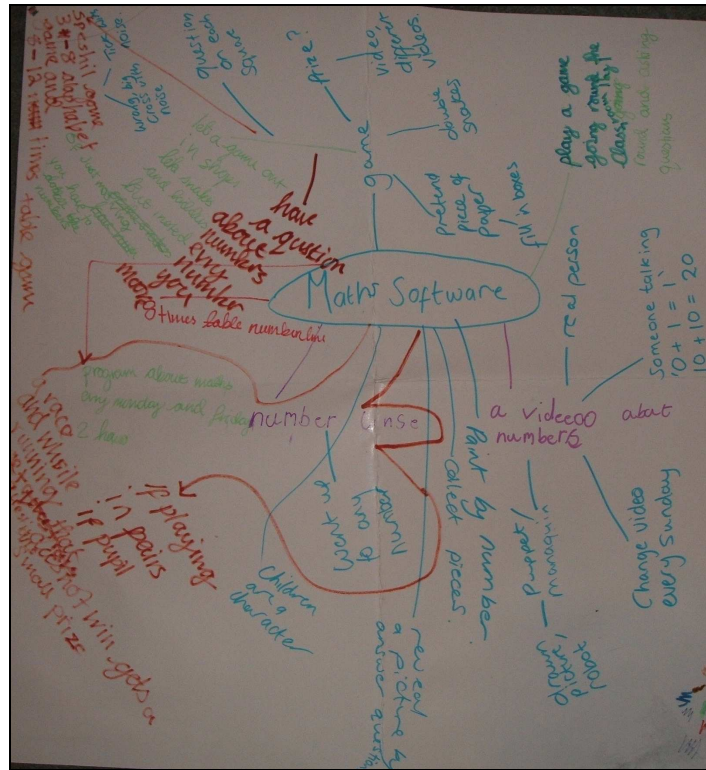


Figure F.1 Spider Diagram of ideas for Maths Software from pilot brainstorming session

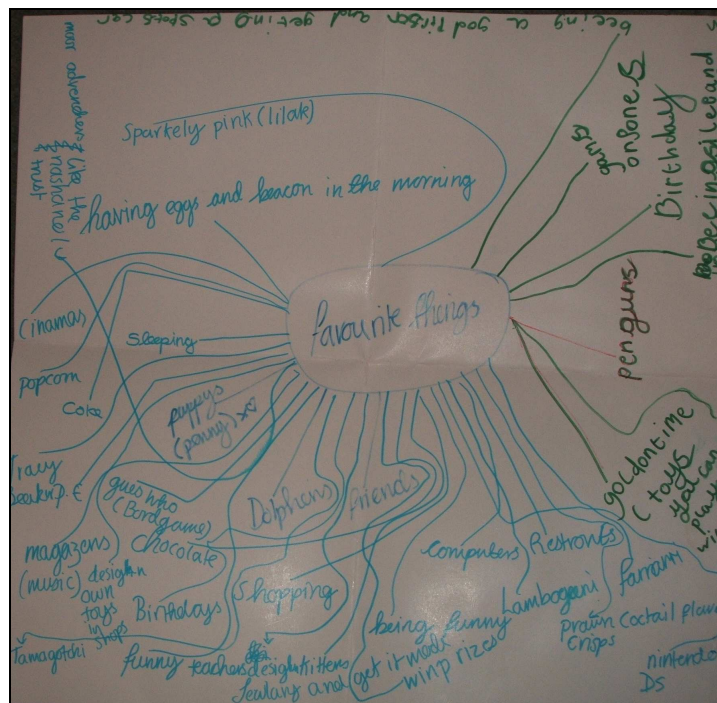


Figure F.2 Spider Diagram of 'favourite things' from pilot brainstorming session

G Brainstorming Session Spider Diagrams

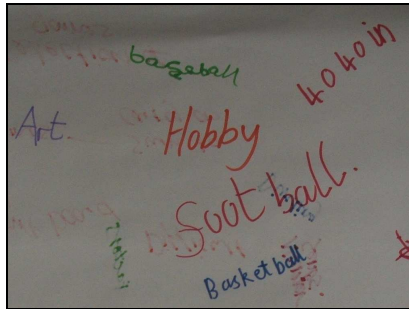


Figure G.1 Spider diagram of favourite hobbies

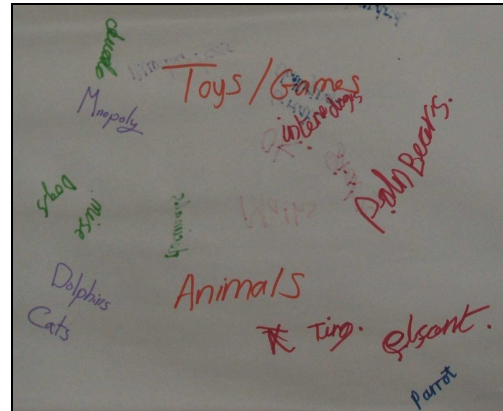


Figure G.2 Spider diagram of favourite toys/games and animals

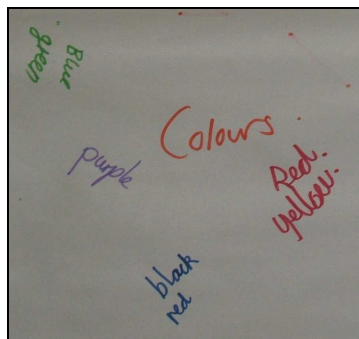


Figure G.3 Spider diagram of favourite colours

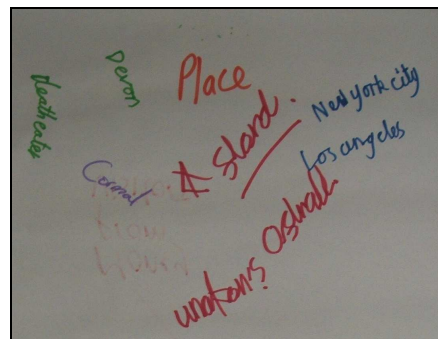


Figure G.4 Spider diagram of favourite places

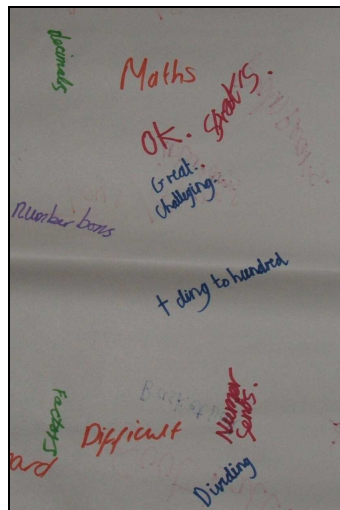


Figure G.5 Spider diagram about opinions on Maths

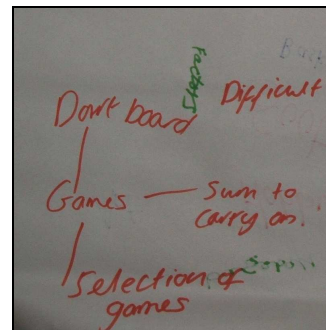


Figure G.6 Spider diagram about ideas for a Maths game

H CARD Session Ordered Cards

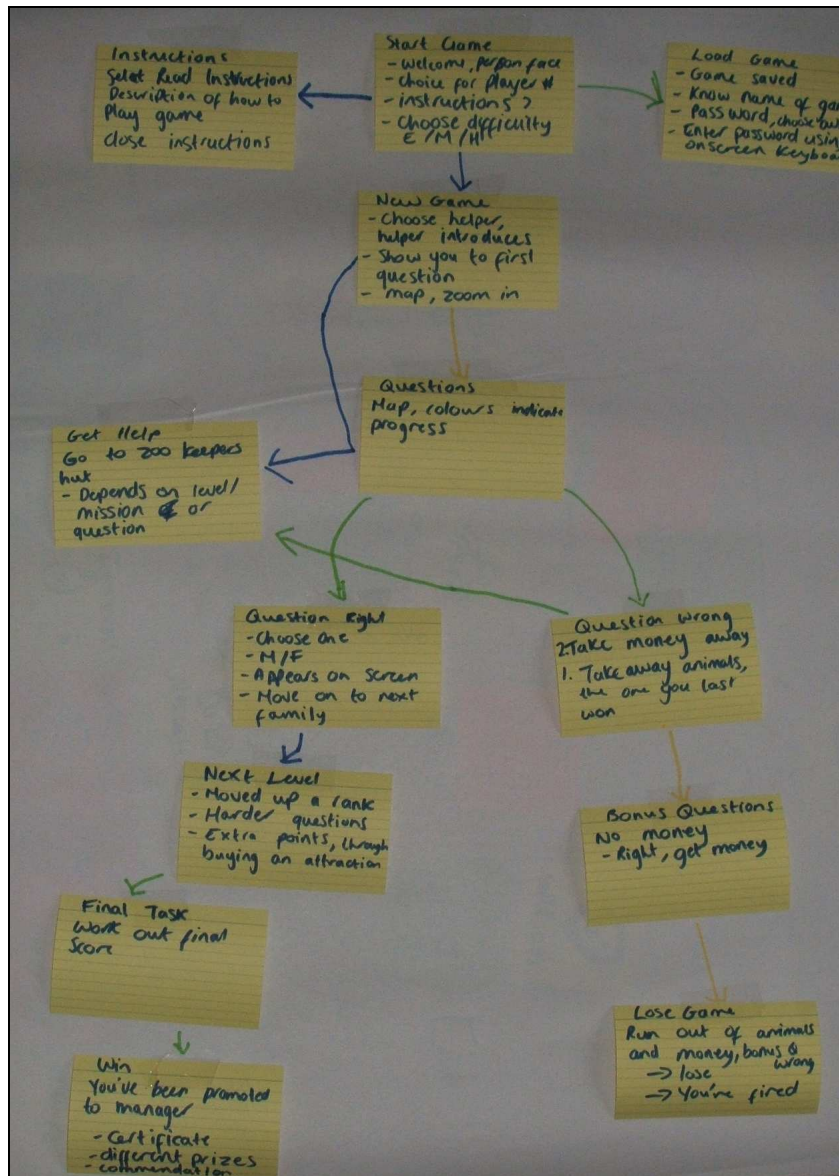


Figure H.1 Ordered cards from CARD session with the design team

I PICTIVE Session Low-Fidelity Prototype



Figure I.1 The Zoo Keepers drawn in the PICTIVE design session



Figure I.2 The Map and Menu drawn in the PICTIVE design session

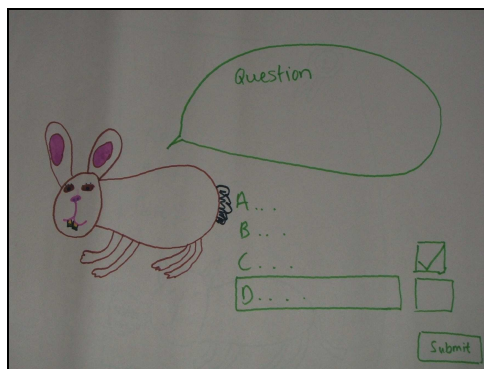


Figure I.3 The Question page drawn in the PICTIVE design session



Figure I.4 The Certificate drawn in the PICTIVE design session

J Initial Game Instructions Text

Maths Zoo Instructions

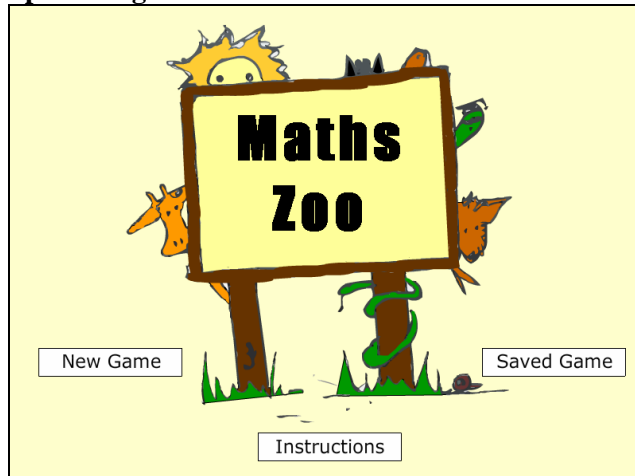
You are the Zoo Keeper at the Maths Zoo and you need to help the zoo by winning animals! There are many zones in the zoo, each has different kinds of animals. You can get to a zone by clicking on the zone from the Zoo Map. When you are in a zone, you will see how many animals of each type you have won and how many you still need to win.

Each zone focuses on maths topics and you will need to answer maths questions correctly to win animals. When an animal has been won, next time you go to that zone, you will see the animal there. If you need help to answer a question, click on 'Help 'and you will be able to do a tutorial on that topic to improve your understanding.

When each of the zones are full with animals, you will have completed the maths zoo and will be able to print a certificate to show your achievement. Good luck!

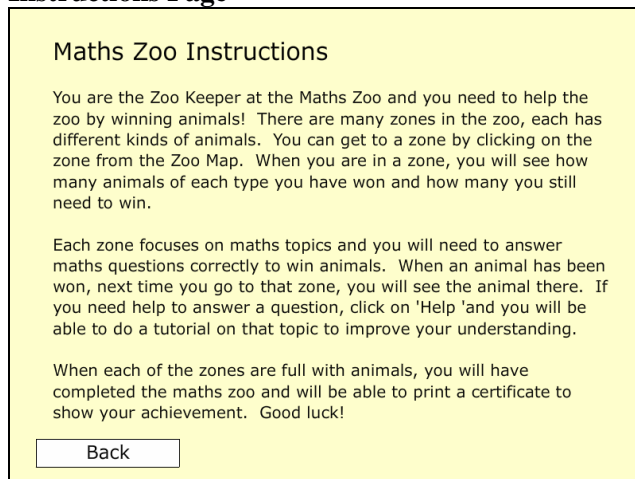
K Maths Zoo Alpha Prototype

Splash Page



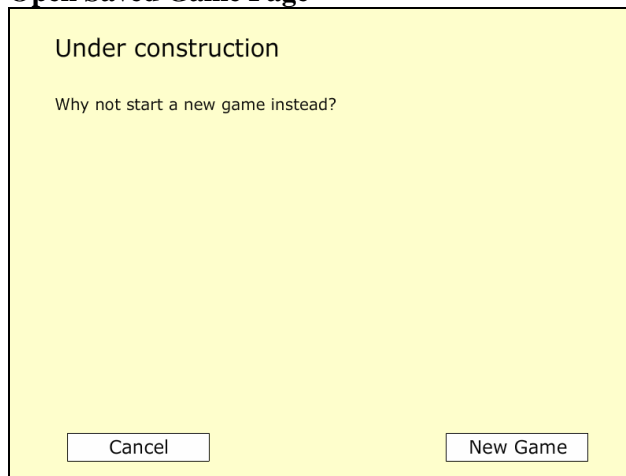
The Splash page is the first that opens when the game is loaded. It consists of the Maths Zoo logo and buttons to start a 'New Game', open a 'Saved Game' and read the 'Instructions'.

Instructions Page



The Instructions consists of a one page written description of how to play the Maths Zoo. The 'Back' button links back to the Splash page.


Open Saved Game Page



The Open Saved Game page has not and will not be implemented in the prototype in view of the high level design decisions (see Section 5.2). However this page has been put in because a full implementation of the game should allow saved games to be opened. The 'Cancel' button links back to the Splash page and the 'New Game' button links to the New Game Options page.

New Game Options

Choose the game options:


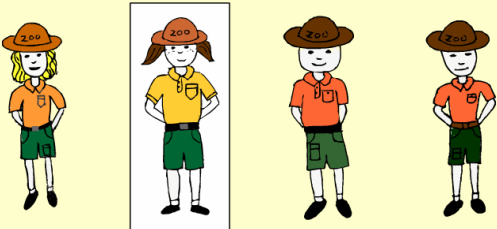


Number of Players	<input type="text" value="One"/>	Two		
School Year	3	4	5	6
Difficulty	Easy	Medium	Hard	

The New Game Options page allows the player to select new game options. Only 'One' player, year '5' and 'Medium' difficulty have been implemented in view of the high level design decisions (see Section 5.2). These available options are in black, whilst the other options are greyed out. When an option is selected, a box appears around it to make the selection obvious. The 'Cancel' button links back to the Splash page and the 'Next' button links to the Choose Zoo Keeper page.

Choose Zoo Keeper

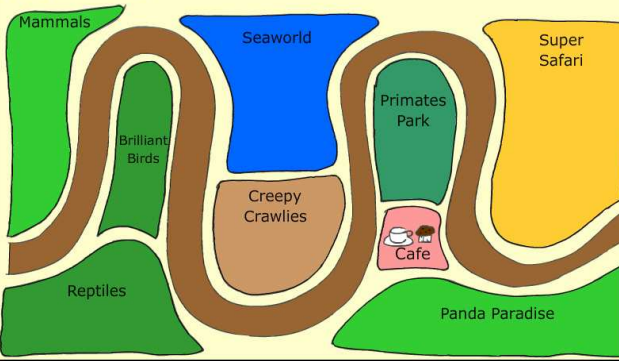
Choose your Zoo Keeper:

The player chooses their zoo keeper for the game by directly clicking on a zoo keeper. When a zoo keeper has been selected a box appears around it to make the selection obvious. It is possible to change the selection. The 'Cancel' button links back to the New Game Options Page and the 'Start' button links to the Map page.

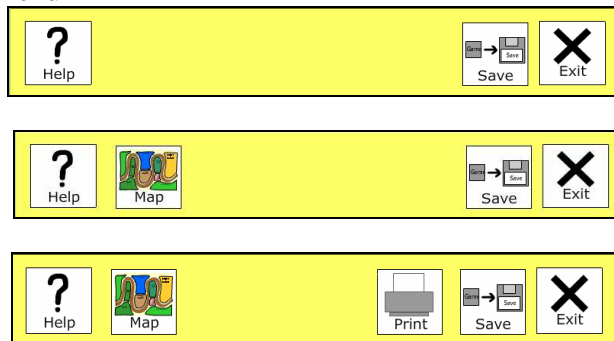
Map

Choose a zone:



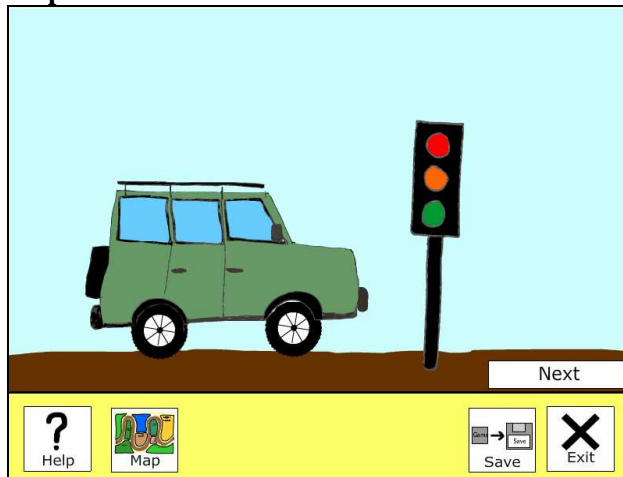
The player chooses a zone by directly clicking on the zone from the map. This then links to the Jeep page, which in turn links to the appropriate zone.

Menu



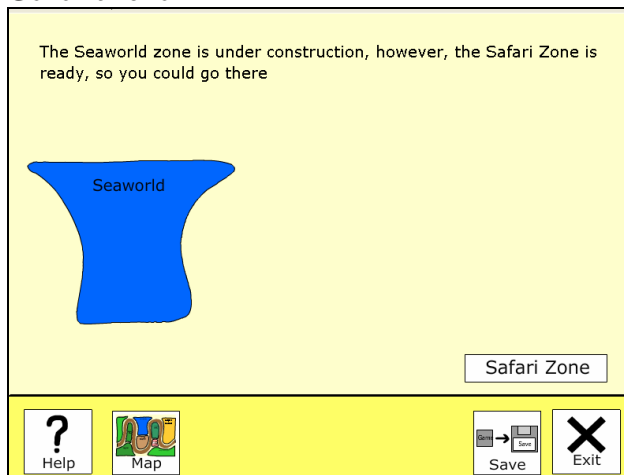
The menu bar has buttons linking to different features and areas of the game; however, the menu bar is not always visible. The menu always has 'Help', 'Save' and 'Exit' buttons. When away from the map, there is a 'Map' button linking back to the Map page. When viewing the certificate, the 'Print' button is shown which when pressed opens the print dialogue box.

Jeep



The Jeep page has an animation of a jeep that moves across the screen until it is stopped by a road item. The 'Next' button links to the page of the selected zone.

Generic zone



Not all zones will be implemented for the purpose of the prototype, so a page for the zone is dynamically customised depending on what zone has been selected.


The page informs the player that the zone is under construction.


There is a 'Safari Zone' button linking to the Safari Zone, which has been implemented.





Safari Zone

Welcome to the Super Safari Zone

Try and win a male and female of each type of animal by answering maths questions



	Male	Female
Giraffe	 Get Animal	Get Animal
Elephant	Get Animal	Get Animal
Lion	Get Animal	Get Animal


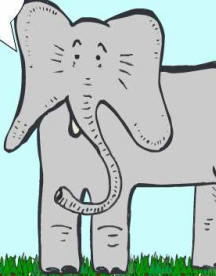
 Help
  Map
  Save
  Exit

The Safari Zone page has a welcome message and instructional text.





There is a table with cells that represent animals in the zone. When an animal still needs to be won, the cell has a 'Get Animal' button that links to a question. When an animal has been won, the animal picture is displayed in the relevant cell.

Question

What is the right answer to the question?

Wrong Answer
 Right Answer
 Wrong Answer
 Wrong Answer

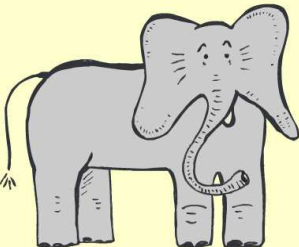
 Help
  Map
  Save
  Exit

The question page displays the Zoo Keeper that was selected at the start of the game and the animal the question is for. The animal has a speech bubble with the question in it.





There are four multiple choice answer buttons. When the right answer button is selected, the Question Right page will open. When any wrong answer button is selected, the Question Wrong page will open.

Question Right page

Well done, you have won a male elephant!



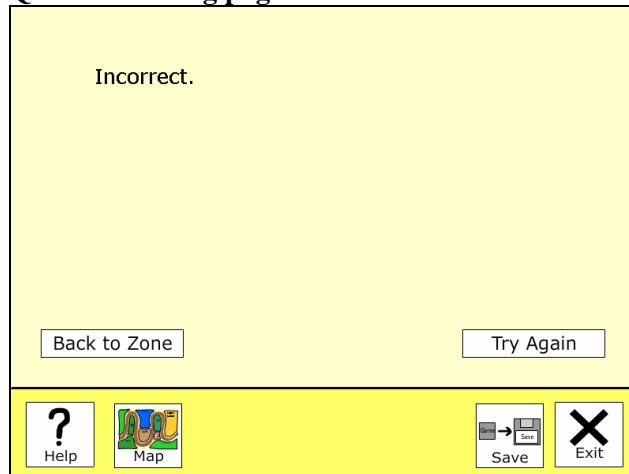
Back to Zone

 Help
  Map
  Save
  Exit

The Question Right page gives a 'well done' praise and notifies as to which animal has been won.

There is a 'Back to Zone' button that links back to the zone.

Question Wrong page



The Question Wrong page notifies that the answer was incorrect. There is a 'Back to Zone' button that links back to the zone and a 'Try Again' button to go back to the same question.

Certificate page

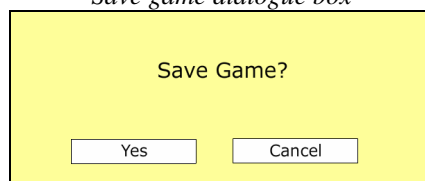


The Certificate page has a picture of scroll-like paper for the certificate. The certificate shows the player's name and the date the certificate was won. There is also a black and white picture of an animal that can be coloured in when printed and a picture of a rosette.

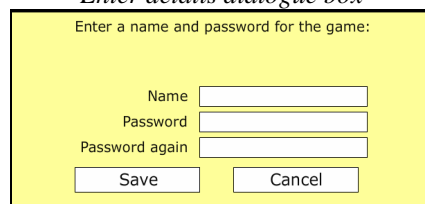
At this stage of the prototype, the functionality to enter the player's name and choose an animal has not been implemented.

Save Game Dialogue Boxes

Save game dialogue box



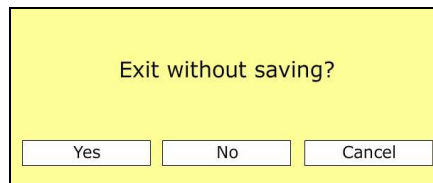
Enter details dialogue box



When the 'Save' button on the menu is pressed, the save game dialogue box comes up to ask the player if they want to save the game. When the 'Yes' button is pressed, the 'Enter Details' dialogue box opens. Pressing the 'Save' button would then save the game, however this has not been implemented.

The 'Cancel' buttons on both dialogue boxes close the dialogue box and returns the player to their previous position in the game.

Exit Game Dialogue Box



When the 'Exit' button is pressed from the menu bar, the Exit Game dialogue box comes up to ask the player if they wish to exit the game without saving. Clicking the 'Yes' button will cause the game to close immediately. Clicking the 'No' button will bring up the Enter Details dialogue box, where the player enters the game details. Clicking 'Cancel' will close the exit dialogue box and return the player to their previous position in the game.

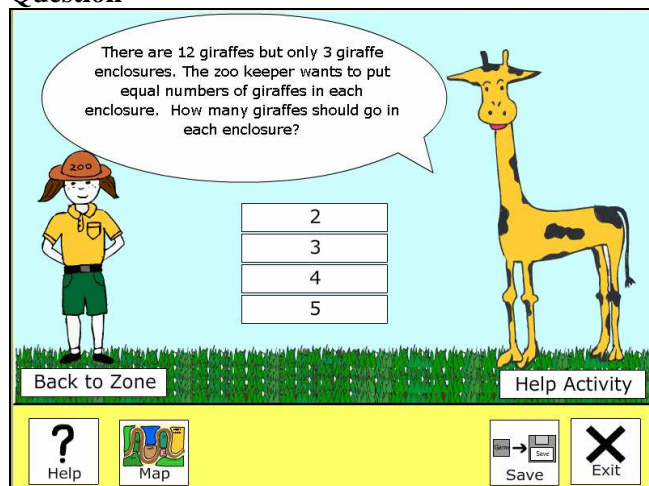
Print Dialogue Box



When the 'Print' button is pressed from the menu bar, the Print dialogue box will appear. Clicking the 'Yes' button should send the certificate to be printed, but this has not yet been implemented. Clicking the 'Cancel' button will close the print pop-up and return the player to their previous position in the game.

Following the initial evaluations with the design team children (see Section 6.2.1), questions and help activities were implemented as was discussed with the children. These additional pages were also evaluated by the HCI expert and are as follows:

Question

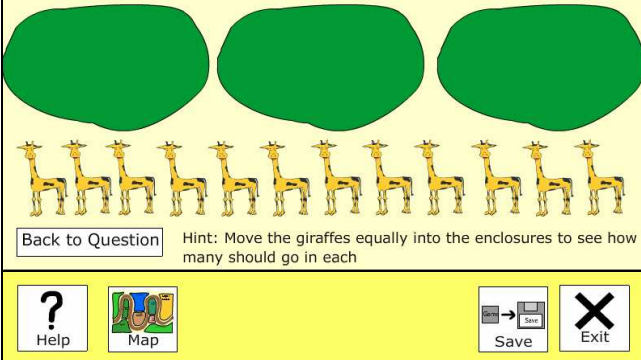


This is an example of how the question page looks. This particular question is for a giraffe; however the other animal questions show the appropriate animal.

Buttons have been added to go 'Back to the Zone' and open the 'Help Activity'.

Giraffe Help Activity

There are 12 giraffes but only 3 giraffe enclosures. The zoo keeper wants to put equal numbers of giraffes in each enclosure. How many giraffes should go in each enclosure?



The interface shows three green oval enclosures at the top. Below them is a row of 12 yellow giraffe icons. At the bottom, there is a yellow bar containing a 'Back to Question' button, a 'Hint: Move the giraffes equally into the enclosures to see how many should go in each', and three icons: a question mark (Help), a map (Map), and a save icon (Save) next to an exit icon (Exit).

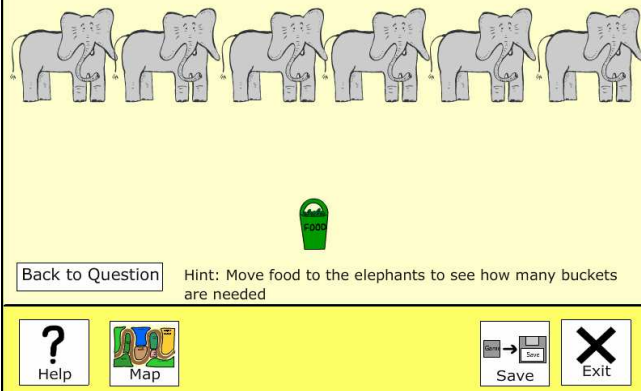
The giraffe help activity shows the appropriate number of fields and giraffes. The giraffes can be moved to the fields by means of drag and drop.

The help activity gives a hint to the player on how to do the help activity.

The help activity has a button to go 'Back to Question'.

Elephant Help Activity

Every day the zoo keeper needs to make sure there is enough food for the elephants. There are 6 elephants, each needs to eat 3 buckets of food every day. How many buckets of food are needed every day for the elephants?



The interface shows a row of 6 grey elephant icons at the top. Below them is a single green bucket icon labeled '1000'. At the bottom, there is a yellow bar containing a 'Back to Question' button, a 'Hint: Move food to the elephants to see how many buckets are needed', and three icons: a question mark (Help), a map (Map), and a save icon (Save) next to an exit icon (Exit).

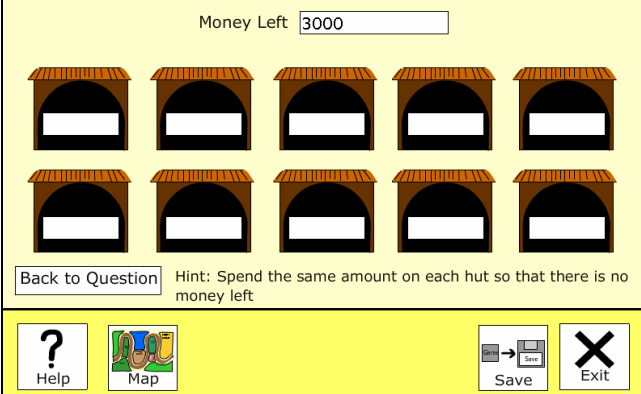
The elephant help activity shows the appropriate number of elephants and a seemingly limitless number of buckets, so that the player can move the necessary number of buckets to the elephants.

The help activity gives a hint to the player on how to do the help activity.

The help activity has a button to go 'Back to Question'.

Lion Help Activity

There is £3000 available to build 10 new identical lion huts, what is the budget for each lion hut?



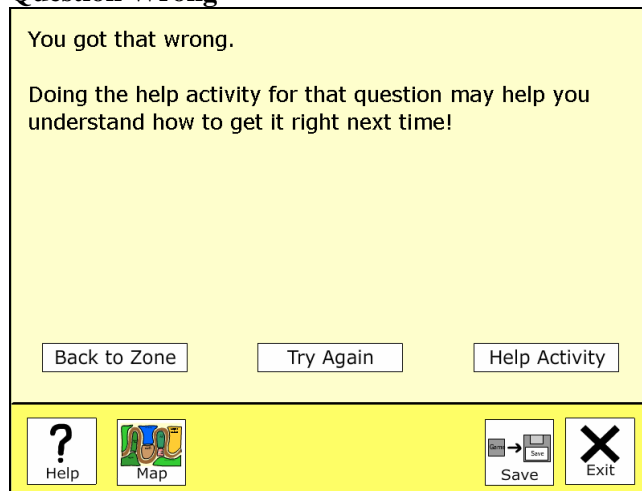
The interface shows a 'Money Left' label followed by an input field containing the value '3000'. Below this is a grid of 10 lion hut icons arranged in two rows of five. At the bottom, there is a yellow bar containing a 'Back to Question' button, a 'Hint: Spend the same amount on each hut so that there is no money left', and three icons: a question mark (Help), a map (Map), and a save icon (Save) next to an exit icon (Exit).

The lion help activity shows 10 lion huts, each with an input field. When a numeric value is inserted into a field, the 'Money Left' value is updated.

The help activity gives a hint to the player on how to do the help activity.

The help activity has a button to go 'Back to Question'.

Question Wrong



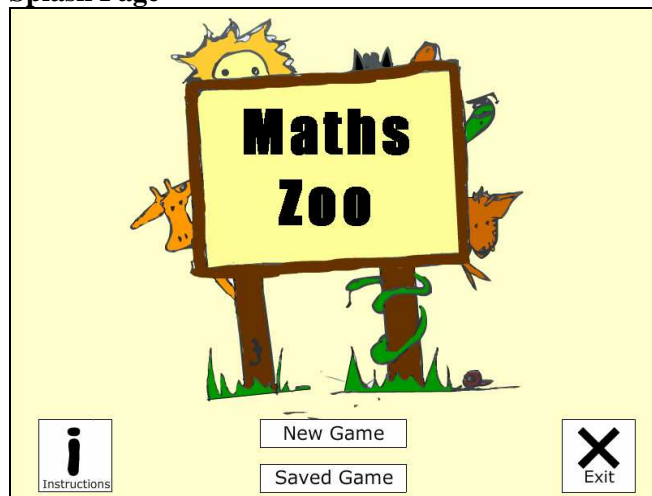
If the question is answered incorrectly, the player is informed of this. The page suggests the player tries the help activity next time then gives options to go 'Back to Zone', 'Try Again' or use the 'Help Activity'.

L Maths Zoo Beta Prototype

Based on the phase one evaluations in the user-centred re-design (see Section 6.2), the Maths Zoo prototype has been changed which has resulted in this Beta.

Pages will only be shown where they have been added to or changed from the previous version; otherwise it can be presumed that they have remained the same as they were in the Alpha prototype.

Splash Page



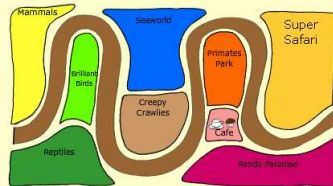
The Splash page has been modified with the addition of an 'Exit' button and an iconic button replacing the previous instructions button. With the extra button it has been necessary to rearrange the buttons.

Instructions

Maths Zoo Instructions

You are the Zoo Keeper at the Maths Zoo. You need to win animals for the zoo by doing maths questions.

There are many zones in the zoo, each with different animals and different types of questions. You can get to a zone from the Zoo Map. This is what the map looks like:



Back

Next

The instructions have been simplified with 'difficult' words being replaced or sentences re-phrased. Shorter sentences have been used and there are more frequent line breaks.

Pictures and icons have been added to aid understanding.

The 'Next' button on the first two pages link to the next page in the instructions, whereas the 'Next' button on the last page links back to the Splash page.

Maths Zoo Instructions

In a zone, you will see how many animals you have won and how many you need to win.

To go to a question from a zone, click on a button that looks like this:

Click here
to win
animal

If you get the question right you will win the animal and win 10 points.

If you get a question wrong, you will not get the animal and will lose 5 points.

Back

Next

The 'Back' button on the first page links back to the Splash page, whereas the 'Back' button on the last two pages link back to the previous page in the instructions.

Maths Zoo Instructions

If you are stuck on a question, you can do an activity to help you work out the answer. To do this click on the button that looks like this:



When you have won every animals in a zone, the zone is complete and you can print a certificate.

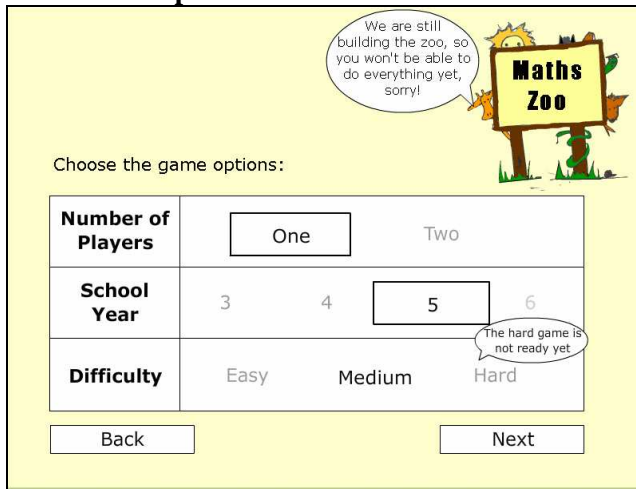
When you have completed every zone, you will have finished the Maths Zoo and you can print another certificate.

Good luck!

Back

Next

New Game Options



We are still building the zoo, so you won't be able to do everything yet, sorry!

Maths Zoo

Choose the game options:

Number of Players	One		Two	
School Year	3	4	5	6
Difficulty	Easy		Medium	Hard

The hard game is not ready yet

Back Next

A speech bubble has been added informing that not all aspects in the zoo are ready yet.

Tool tips have been added to the unavailable options.

The 'Cancel' button has been replaced with to a 'Back' button, which links back to the Splash page.

New Game Details



Maths Zoo

Enter information for your new game:

Your Name:

Game Name:

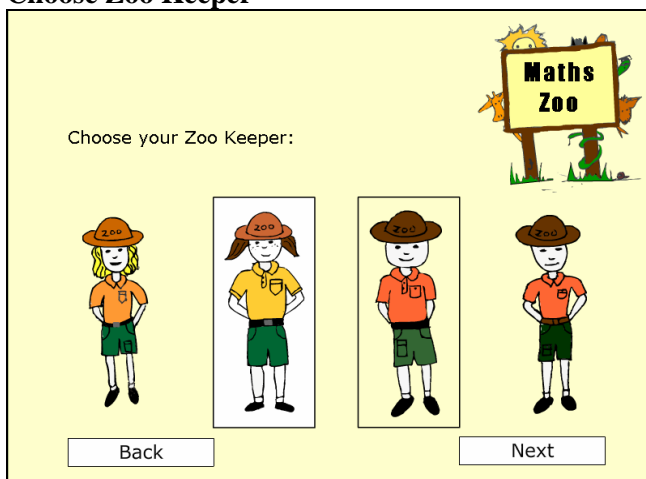
Game Password:

Game Again:

Back Next


The Game Details page is new in this version; it has been added so that the player can enter their name which is later used on their certificate. The other information is captured so that the game can be saved at a later point, however, game saving will not be implemented in the prototype.

Choose Zoo Keeper





Maths Zoo


Choose your Zoo Keeper:



Back



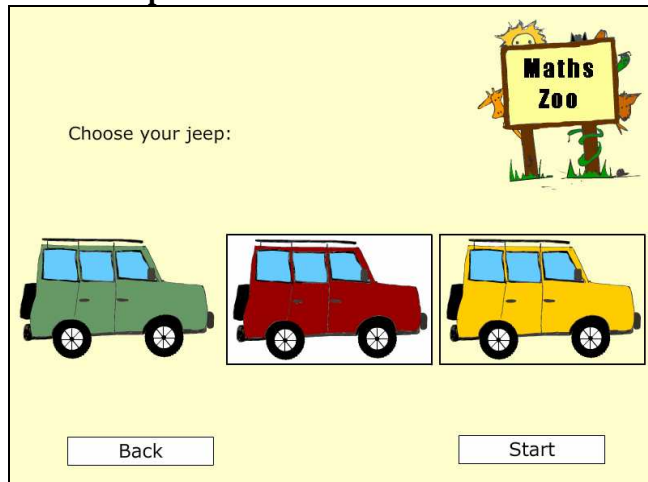




Next

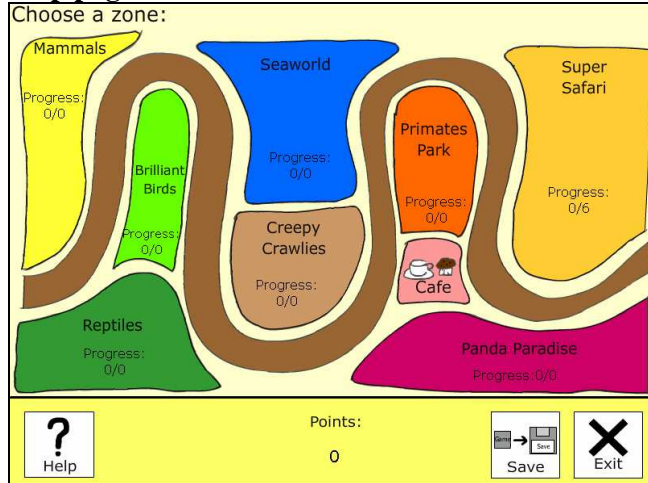
The roll over effect has been changed to show a difference between the selected zoo keeper and the zoo keeper the mouse is hovering over. A white rectangle around a zoo keeper indicates that it has been selected.

Choose Jeep



The choose jeep page is new in this version and has been added out of request by the children involved in evaluations because they wanted to be able to choose the colour of their jeep. Again, a white rectangle indicates which jeep has been selected.

Map page

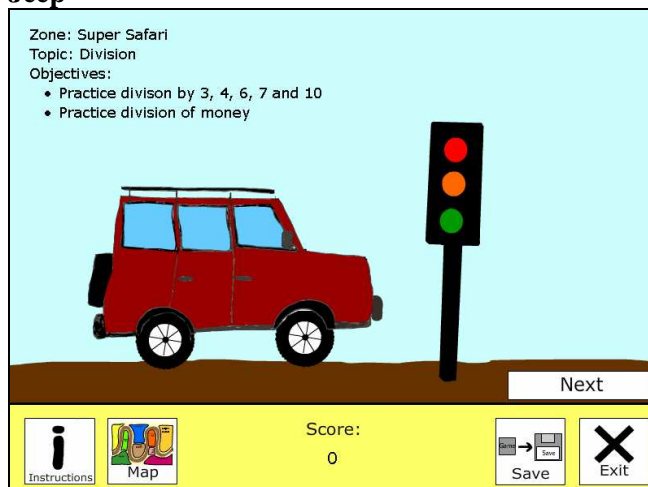


The colours of the zones have been changed to use brighter colours, at the request of the children involved in evaluations.

Each zone now shows the progress i.e. 0/8 for the Super Safari zone.

The game points can now be seen in the middle of the menu bar.

Jeep



The jeep page now reflects the jeep chosen at the start of the game.

The zone topic and objectives have been added so that the player is aware of the mathematical content and the purpose of the zone.

Super Safari Zone

Welcome to the Super Safari Zone

Try and win a male and female of each type of animal by answering maths questions

Super Safari

	Male	Female
Giraffe	Click here to win animal	Click here to win animal
Elephant	Click here to win animal	Click here to win animal
Lion	Click here to win animal	Click here to win animal

Points: 0

Help Map Save Exit

The instructional text now comes from the zoo keeper, to make it more obvious.

The buttons in the table have been changed from 'Get Animal' to 'Click here to win animal'.

Question page

Question:
There is £3000 available to build 10 new identical lion huts, what is the budget for each lion hut?

£3000
£300
£30
£3

Points: 0

Help Map Zone Save Exit

The question page has been changed so that the navigation buttons are no longer in the content page of the page. Instead, navigation occurs from the menu with the 'Zone' button linking back to the zone and the 'Help' button linking to the help activity.

Giraffe Help Activity

Question:
There are 12 giraffes but only 3 giraffe fields. The zoo keeper wants to put equal numbers of giraffes in each field. How many giraffes should go in each field?

Move the giraffes equally into the enclosures to see how many should go in each

Back to Question

Points: 10

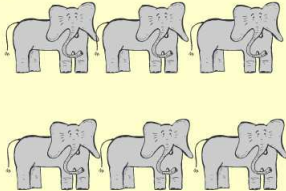

Help Map Zone Save Exit

The hint is now given by the zoo keeper so that it is more obvious.






Elephant Help Activity

Question:
Every day the zoo keeper needs to make sure there is enough food for the elephants. There are 6 elephants, each needs to eat 3 buckets of food every day. How many buckets of food are needed every day for the elephants?

Move food to the elephants to see how many buckets are needed

Back to Question




 Points: 10
 


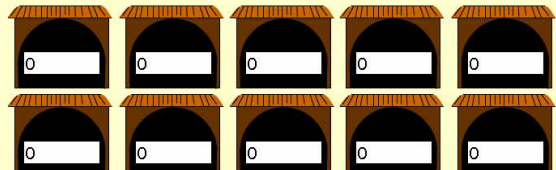

The layout of the page has been restructured to allow for the addition of the zoo keeper and help text.

Lion Help Activity

Question:
There is £3000 available to build 10 new identical lion huts, what is the budget for each lion hut?






Spend the same amount of money on each hut so that there is no money left

Money Available

Back to Question

Enter Amount Enter





 Points: 10
 








The lion activity has been changed so that only one value needs to be entered. This value is then populated across the fields on the lion huts, with the money available value also updated.






Completed Safari Zone

Welcome to the Super Safari Zone

You have completed the Safari zone and can now print a Maths Zoo Certificate!



	Male	Female
Giraffe		
Elephant		
Lion		



 Points: 60
 



On completion of the zone, a 'Certificate' button now appears in the menu, linking to the certificate.

The zoo keeper also informs that the zone has been completed and reminds the player to get their certificate.

Save game dialogue boxes

Save the game?

Yes No

Game Saved!

OK

The save process has been changed so the user enters their details and game details at the start of a game. The save game dialogue boxes have been changed to reflect this.

Exit game dialogue box

Leave the game?

Yes No

The wording on the exit game dialogue box has been changed. If the player clicks 'No', the exit is cancelled but if they click 'Yes' they will be prompted to save the game before exiting.

On-screen keyboard

Capitals Capitals off

A	B	C	D	E	F	G	H	I	1	2	3
J	K	L	M	N	O	P	Q	R	4	5	6
S	T	U	V	W	X	Y	Z		7	8	9
Delete	Space								0		

Text entered: Enter Cancel

The on-screen keyboard has now been implemented for the purpose of the interactive whiteboard evaluations.

Super Safari Zone

Welcome to the Super Safari zone

Try and win a male and female of each type of animal by answering maths questions

	Giraffe	Zebra	Elephant	Lion
Male	Click here to win animal	Click here to win animal	Click here to win animal	Click here to win animal
Female	Click here to win animal	Click here to win animal	Click here to win animal	Click here to win animal

Score: 0

Instructions Map Save Exit

After the pair evaluations in phase 2 of re-design (see Section 6.3.1), Zebras were added to the zone on request of the children to have more animals and questions.

Question

Question:
There are 7 zebra stables and the zoo keeper has 21 bundles of hay to share between them, how many bundles of hay should go to each stable?

Use help if you are stuck!

4
2
5
3

? Help Map Zone Score: 0 Save Exit

This zebra question was implemented.

Also the zoo keeper now reminds the player to use 'Help' if they are stuck.

Zebra Activity

Question:
There are 7 zebra stables and the zoo keeper has 21 bundles of hay to share between them, how many bundles of hay should go to each stable?

Move the hay to the zebras so that they have the same number each

Q Question Map Zone Score: 20 Save Exit

The zebra help activity features the appropriate number of zebras and hay bundles. The hay bundles can be moved to the zebras by drag and drop.

When in the help activity, the Question is now linked back to from the menu.

Elephant Activity page

Question:
The zoo keeper has 18 buckets of food for 6 elephants, how many buckets of food should he give to each elephant so that they have the same?

Move food to the elephants to see how many buckets are needed

Q Question Map Zone Score: 40 Save Exit

The elephant question and activity was changed from a multiplication to a division question, to keep inline with the zone's topic of division. This has resulted in some restructuring of the layout and the appropriate number of buckets being shown, rather than a seemingly limitless number.

Lion Help Activity

Question:
There is £8760 available to build 10 new identical lion huts, what is the budget for each lion hut?

Money Available:

Spend the same amount of money on each hut so that the money available is 0, use the box below to enter an amount

Enter Amount:

Score: 55

Question Map Zone Save Exit

A button has been added to access the on-screen keyboard.

Question Wrong

That was the wrong answer.

You probably gave the elephants different amounts of food buckets, remember they need to have the number of buckets of food.

Doing the help activity for that question may help you get it right next time!

Score: 35

The Question Wrong page now gives advice as to why the question may have been answered incorrectly.

Certificate

Choose an animal for your certificate:

Maths Zoo Certificate

Name: Kat

Zone: Super Safari
Topic: Division
Score: 75 out of 80

Score: 75

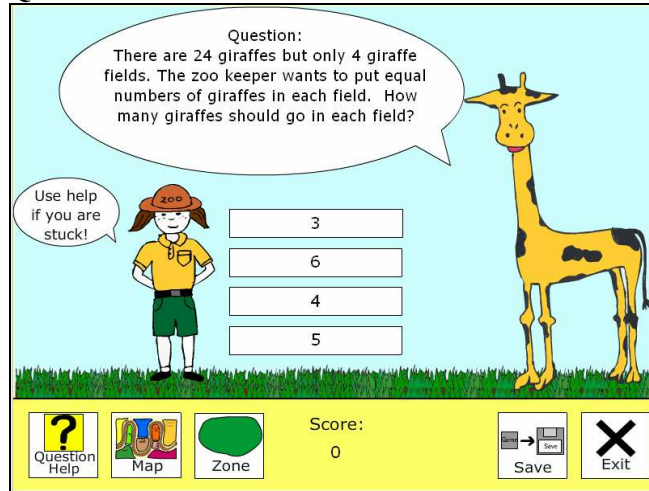
The functionality to choose the animal for the certificate has now been implemented. A white square indicates which animal has been selected, but it is possible to change the choice.

M Maths Zoo 1.0 Prototype

Based on the phase two evaluations in the user-centred re-design (see Section 6.3), the Maths Zoo prototype has been changed which has resulted in this final version, Maths Zoo 1.0.

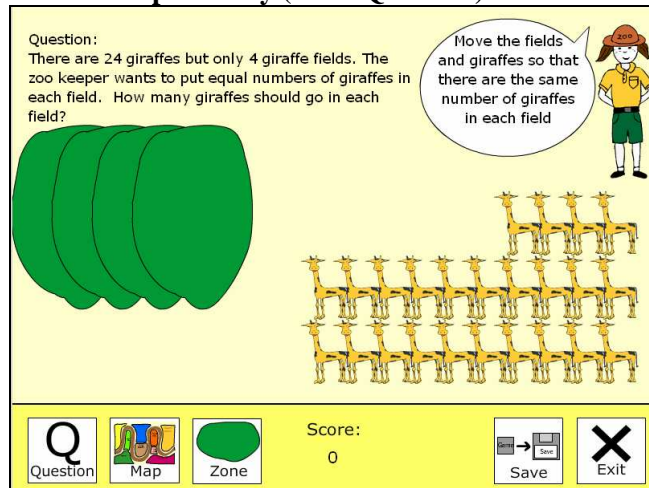
Pages will only be shown where they have been added to or changed from the previous version; otherwise it can be presumed that they have remained the same as they were in the Beta prototype.

Question



The 'Help' button in the menu has been changed to a 'Question Help' button, as to make its purpose more obvious.

Giraffe Help Activity (Male Question)

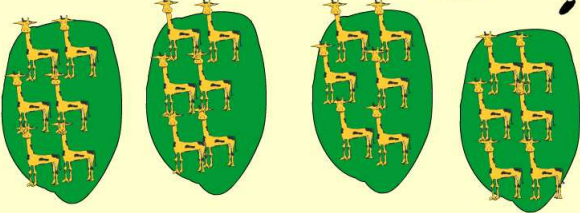


As more questions have been implemented, the number of fields and giraffes are now dynamically created; therefore they are lined up in rows.

In this question the number of fields and giraffes match the numbers in the question.

Question:
There are 24 giraffes but only 4 giraffe fields. The zoo keeper wants to put equal numbers of giraffes in each field. How many giraffes should go in each field?

Move the fields and giraffes so that there are the same number of giraffes in each field



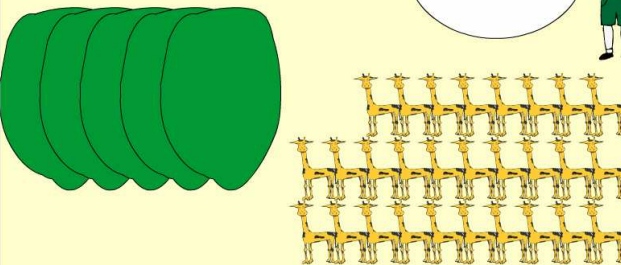
Question Map Zone Score: 0 Save Exit

As can be seen, both the fields and giraffes can be moved. Six giraffes can be put in each field so that there are equal numbers in each.

Giraffe Help Activity (Male Question)

Question:
If there are 28 giraffes and only 9 giraffes can go in a field, how many fields are needed?

Try and fill each field with 9 giraffes to see how many fields are needed

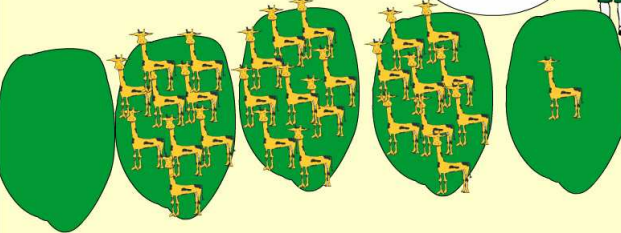


Question Map Zone Score: -20 Save Exit

This giraffe question is slightly different; there are more fields than needed because the player needs to work out the number of fields needed based on the number of giraffes and capacity of each field.

Question:
If there are 28 giraffes and only 9 giraffes can go in a field, how many fields are needed?

Try and fill each field with 9 giraffes to see how many fields are needed



Question Map Zone Score: -20 Save Exit

As can be seen, Four fields are needed, even though one field only has one giraffe in.

Zebra Help Activity (Male Question)

Question:
There are 7 zebra stables and the zoo keeper has 35 bundles of hay to share between them, how many bundles of hay should go to each zebra stable?

Move the zebras and hay so that each zebra has the same number of hay bundles

Question

Map

Zone

Score: 0

Save

Exit

The zebras and hay bundles are now lined up in rows as they are dynamically created. The numbers of zebras and hay bundles match that in the question.

Question:
There are 7 zebra stables and the zoo keeper has 35 bundles of hay to share between them, how many bundles of hay should go to each zebra stable?

Move the zebras and hay so that each zebra has the same number of hay bundles

Question

Map

Zone

Score: 0

Save

Exit

As can be seen, the hay bundles can be moved to the zebras. Each zebra should be given five hay bundles each.

Zebra Help Activity (Female Question)

Question:
There are 6 zebras and each zebra needs 7 bundles of hay a week. If there are 40 bundles of hay, how many more bundles of hay does the zoo keeper need to get?

Try to move 7 bundles of hay to each zebra, to see how many more bundles are needed for every zebra to have 7 bundles of hay

Question

Map

Zone

Score: 0

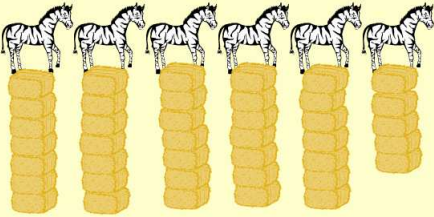
Save

Exit

In this zebra activity, the player has to work out how many more bundles of hay are needed.

Question:
There are 6 zebras and each zebra needs 7 bundles of hay a week. If there are 40 bundles of hay, how many more bundles of hay does the zoo keeper need to get?

Try to move 7 bundles of hay to each zebra, to see how many more bundles are needed for every zebra to have 7 bundles of hay



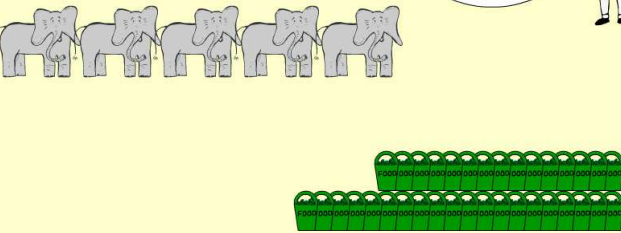
Question Map Zone Score: 0 Save Exit

As can be seen, two more hay bundles are needed.

Elephant Help Activity (Male Question)

Question:
The zoo keeper has 35 buckets of food for 5 elephants, how many buckets of food should he give to each elephant so that they have the same?

Move the buckets of food to the elephants so that they each have the same number

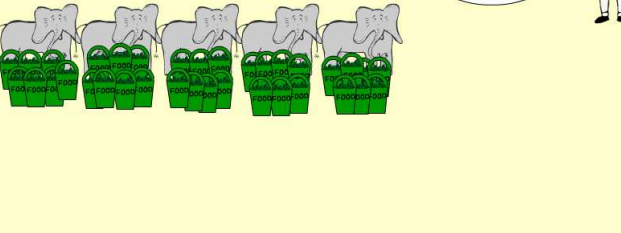


Question Map Zone Score: 0 Save Exit

This elephant help activity has the same number of elephants and food buckets as in the question.

Question:
The zoo keeper has 35 buckets of food for 5 elephants, how many buckets of food should he give to each elephant so that they have the same?

Move the buckets of food to the elephants so that they each have the same number



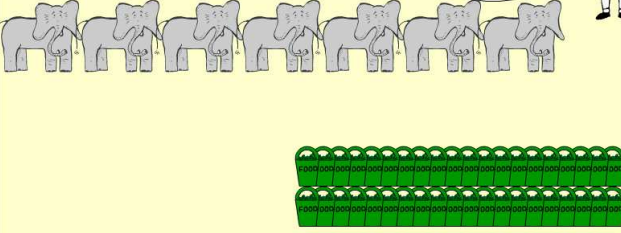
Question Map Zone Score: 0 Save Exit

As can be seen, each elephant can have 7 buckets of food each.

Elephant Help Activity (Female Question)

Question:
If there are 40 buckets of food and an elephant needs to eat 7 buckets of food, how many elephants will this feed?

Try to move 7 buckets of food to each elephant to see how many elephants can be fed all 7 buckets of food

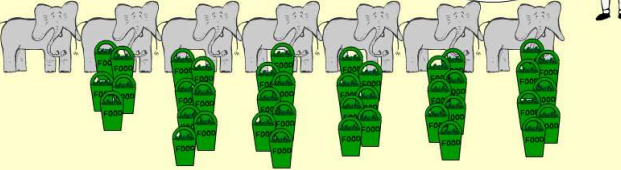


Question Map Zone Score: 0 Save Exit

In this help activity the player has to work out how many elephants can be fed based on the number of food buckets available.

Question:
If there are 40 buckets of food and an elephant needs to eat 7 buckets of food, how many elephants will this feed?

Try to move 7 buckets of food to each elephant to see how many elephants can be fed all 7 buckets of food



Question Map Zone Score: 0 Save Exit

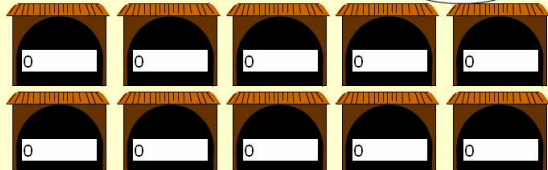
As can be seen, only five elephants can be fed all seven food buckets.


Lion Help Activity (Male Question)

Question:
There is £560 available to build 10 new identical lion huts, what is the budget for each lion hut?

Money Available

Spend the same amount of money on each hut so that the money available is 0, use the box below to enter an amount



Enter Amount: Enter 

Question Map Zone Score: 0 Save Exit

This help activity is as before.

Question:
There is £560 available to build 10 new identical lion huts, what is the budget for each lion hut?

Money Available

Spend the same amount of money on each hut so that the money available is 0, use the box below to enter an amount

56	56	56	56	56
56	56	56	56	56

Enter Amount: Enter

Question Map Zone Score: 0 Save Exit

As can be seen, the player needs to spend £56 on each lion hut.

Lion Help Activity (Female Question)

Question:
The zoo keeper starts work at midday and needs to visit 16 animals in 8 hours. If the zoo keeper spends the same amount of time with each animal, how long does the zoo keeper spend with each animal?

Move the animals to the timeline to work out how long the zoo keeper spends with each animal

12 1 2 3 4 5 6 7 8

Question Map Zone Score: 0 Save Exit

With this lion help activity, the player needs to work out how long the zoo keeper should spend with each animal, by moving the animals up to the time line.

Question:
The zoo keeper starts work at midday and needs to visit 16 animals in 8 hours. If the zoo keeper spends the same amount of time with each animal, how long does the zoo keeper spend with each animal?

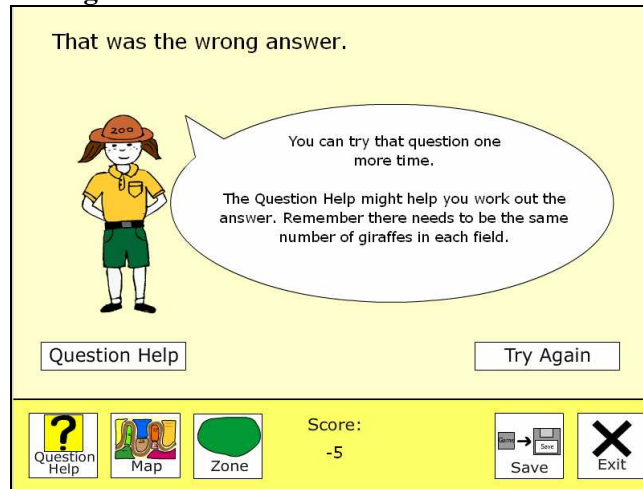
Move the animals to the timeline to work out how long the zoo keeper spends with each animal

12 1 2 3 4 5 6 7 8

Question Map Zone Score: 0 Save Exit

As can be seen, the zoo keeper should spend half an hour with each animal.

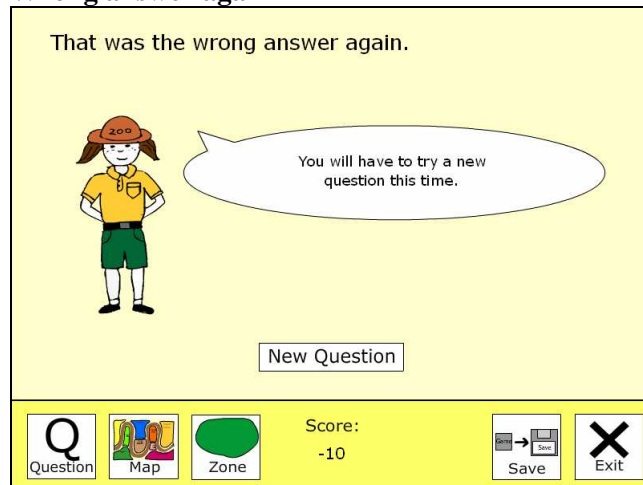
Wrong Answer



The Wrong Answer page has been changed; it has been made friendlier with the addition of the zoo keeper giving the advice.

The zoo keeper also tells the player how many more tries they can have.

Wrong answer again



If the question has been answered wrong again, the player is informed of this and has to try a new question.

N Full Listing of Maths Zoo Questions

Animal	Sex	Question	Multiple Choice	Correct
Giraffe	M	There are 24 giraffes but only 4 giraffe fields. The zoo keeper wants to put equal numbers of giraffes in each field. How many giraffes should go in each field?	3, 6, 4, 5	6
		There are 12 giraffes but only 3 giraffe fields. The zoo keeper wants to put equal numbers of giraffes in each field. How many giraffes should go in each field?	2, 3, 4, 5	4
	F	If there are 28 giraffes and only 9 giraffes can go in a field, how many fields are needed?	2, 3, 4, 5	4
		If there are 29 giraffes and only 6 giraffes can go in a field, how many fields are needed?	5, 3, 6, 4	5
Zebra	M	There are 7 zebra stables and the zoo keeper has 35 bundles of hay to share between them, how many bundles of hay should go to each zebra stable?	4, 5, 2, 3	5
		There are 8 zebra stables and the zoo keeper has 32 bundles of hay to share between them, how many bundles of hay should go to each zebra stable?	4, 2, 5, 3	4
	F	There are 5 zebras and each zebra needs 7 bundles of hay a week. If there are 30 bundles of hay, how many more bundles of hay does the zoo keeper need to get?	2, 3, 4, 5	5
		There are 6 zebras and each zebra needs 7 bundles of hay a week. If there are 40 bundles of hay, how many more bundles of hay does the zoo keeper need to get?	1, 2, 3, 4	2
Elephant	M	The zoo keeper has 24 buckets of food for 8 elephants, how many buckets of food should he give to each elephant so that they have the same?	2, 3, 4, 5	3
		The zoo keeper has 35 buckets of food for 5 elephants, how many buckets of food should he give to each elephant so that they have the same?	5, 6, 7, 8	7
	F	If there are 30 buckets of food and an elephant needs to eat 9 buckets of food, how many elephants will this feed?	5, 4, 3, 2	3
		The zoo keeper has 35 buckets of food for 5 elephants, how many buckets of food should he give to each elephant so that they have the same?	5, 6, 7, 8	
Lion	M	There is £560 available to build 10 new identical lion huts, what is the budget for each lion hut?	£56, £57, £50, £60	
		There is £8760 available to build 10 new identical lion huts, what is the budget for each lion hut?	£860, £870, £867, £876	
	F	The zoo keeper starts work at midday and needs to visit 16 animals in 4 hours. If the zoo keeper spends the same amount of time with each animal, how long does the zoo keeper spend with each animal?	One hour, Half an hour, Two hours, Quarter of an hour	Quarter of an hour
		The zoo keeper starts work at midday and needs to visit 16 animals in 8 hours. If the zoo keeper spends the same amount of time with each animal, how long does the zoo keeper spend with each animal?	One hour, Half an hour, Two hours, Quarter of an hour	Half an hour

O Test Case Sample

The Maths Zoo game is extensive in size and has many paths through it. A sample of test cases will be shown that mimic a typical path a player would take to win their first animal.

Test ID	Test Description	Expected Result	Actual Result	Pass/Fail
1	Open 'mathszoo.exe'	The Splash page will open	As expected	PASS
2	Press the 'Instructions' button	The first page of the instructions will open	As expected	PASS
3	Press the 'Next' button	The second page of the instructions will open	As expected	PASS
4	Press the 'Close' button	The Splash page will open	As expected	PASS
5	Press the 'New Game' button	The Game Options page will open	As expected	PASS
6	Press the 'Next' button	A message will appear to inform that school year and difficulty have not been selected	As expected	PASS
7	Select year '5' and 'medium' difficulty then press the 'Next' button	The Game Details page will open	As expected	PASS
8	Press the 'Next' button	A message will appear to inform that the player has not entered their name or a game name	As expected	PASS
9	Enter 'Katherine' in the 'Your Name field' using a hardware keyboard	'Katherine' will appear in the 'Your Name' field	As expected	PASS
10	Enter 'Game1' in the 'Game Name' field using the on-screen keyboard, then press the 'Enter' button on the onscreen keyboard	As each button on the on-screen keyboard is pressed, the associated character will appear in the 'Text entered' field on the on-screen keyboard. When 'Enter' is pressed 'Game1' will appear in the 'Game Name' field	As expected	PASS
11	Press the 'Next' button	The Choose Zoo Keeper page will open	As expected	PASS
12	Press the 'Next' button	A message will appear to inform that a zoo keeper has not been selected	As expected	PASS
13	Choose a zoo keeper by clicking on one	A white box will appear around the selected zoo keeper	As expected	PASS
14	Click on a different zoo keeper to change the selection	The white box will be removed from around the first selection and will appear around the new selection	As expected	PASS
15	Click the 'Next' button	The Choose Jeep page will open	As expected	PASS
16	Choose a Jeep by clicking on one	A white box will appear around the selected Jeep	As expected	PASS

Test ID	Test Description	Expected Result	Actual Result	Pass/Fail
17	Click on a different Jeep to change the selection	The white box will be removed from around the first selection and will appear around the new selection	As expected	PASS
18	Click the 'Next' button	The Map page will open	As expected	PASS
19	Click on the 'Mammals' zone	The Jeep page will open showing: Zone: Mammals, Topic: ? Objectives ?	As expected	PASS
20	Click the 'Next' button	The Zone page will open informing that the Mammals zone is under construction	As expected	PASS
21	Click the 'Map' button in the menu	The Map page will open	As expected	PASS
22	Click on the 'Super Safari' zone	The Jeep page will open showing: Zone: Super Safari, Topic: Division and listing several objectives	As expected	PASS
23	Click the 'Next' button	The Super Safari zone will open, no animals will have been won yet	As expected	PASS
24	Click the 'Click here to win animal button' for a male giraffe	The Question page will open showing a giraffe asking a question	As expected	PASS
25	Click on a wrong answer button	The question wrong page will open, informing that the answer was wrong, that the player can try one more time and giving a hint as to how it could be worked out. The score will show -5	As expected	PASS
26	Click the 'Try Again' button	The question that was seen before will open again	As expected	PASS
27	Click the 'Question Help' button in the menu	The giraffe help activity page will open. The number of fields and giraffes shown will match the relevant numbers in the question	As expected	PASS
28	Move each of the fields and giraffes to work out the answer	Each field and giraffe will move by drag and drop, equal numbers of giraffes will fit in each field	As expected	PASS
29	Click the 'Question' button in the menu	The question that was seen before will open again	As expected	PASS
30	Click on the right answer button	The Question Right page will open and inform that the question is right and that a male giraffe has been won. The score will show 5	As expected	PASS
31	Click on the 'Zone' button in the menu	The super safari zone will open, a giraffe will be shown in the male giraffe table field	As expected	PASS

P Requirements Verification

Each requirement from the four categories of requirements will be analysed to establish to what extent they have been met by the Maths Zoo 1.0 prototype. Based on this analysis each will be given a status from one of three categories which will now be defined:

- **Met** - All aspects of the requirement have been met
- **Partial** - Some aspects of the requirement have been met
- **Unmet** - No aspects of the requirement have been met

P.1 Functional requirements

Req.	Description	Analysis	Status
1	The system shall provide the user with mathematical instruction linked to the key stage two mathematics national curriculum	The division content fits in the 'Number and algebra' area of the key stage two national curriculum. However, due to high-level design decisions (see Section 5.2) there is only content for year 5 and division, but the infrastructure to put this in place has mostly been implemented	Partial
2	The system should provide a suitably enjoyable learning experience	User experience questionnaires indicate that children enjoy playing the game and that it makes learning fun (see Section 8.4)	Met
3	The system shall provide the user with clear objectives about the purpose of the tasks to complete	The Jeep pages states the topic and objectives for the zone	Met
4	The system should provide a suitably challenging learning experience	The learning evaluations indicate that for year 5 children, the content is challenging for most children in the low-ability set, some children in the middle-ability set and few children in the high-ability set. However, one teachers thought the female lion question (a time question) was suitable for the high ability set	Partial
5	The system shall provide the user with opportunity to practice	The help activities allow for practice. Also, as there can be two attempts at each question, this allows for practice	Met
6	The system shall clearly show the user's progress in task completion	Each zone on the map shows the number of animals that have been won out of the total number of animals. In the zone, there is a pictorial representation of the animals to indicate which ones have been won. The score is visible at all times in the menu	Met
7	The system should allow the user to save their progress in the system and resume use of the system at this saved point	This has not been implemented due to a high level design decisions (see Section 5.2). However, the front end for saving has been implemented	Partial

8	The system shall provide constructive feedback that guides the user away from making the same mistakes again, avoiding negative words	When a question is answered incorrectly, the zoo keeper gives constructive advice and hints as to how the user can work out the answer next time	Met
9	The system shall allow the user to 'try again' for a limited number of times on incorrect completion of a task	Each question can be tried twice before the user is presented with a new question	Met
10	The system shall provide visual learning information	The system has a GUI, the help activities use visual representations of objects to aid the player's understanding in concrete terms	Met
11	The system shall provide relevant learning assistance to the user through an interface persona	The zoo keeper persona gives appropriate advice in each help activity	Met
12	The system should provide usability instructions that are accessible at any point in the system	The instructions can be accessed from the Splash screen. Then when the player is choosing the options for a new game the instructions cannot be accessed because the menu is not visible. When the menu is visible the instructions can be access at any time except when in the question or help activity as the icon is replaced with a button to the help activity or back to the question	Partial
13	The system shall provide reward to the user on correct completion of learning tasks	The player wins an animal and 10 points for every correctly answered question	Met
14	The system should give rewards valued by the user	User experience questionnaires indicate that the children like the zoo theme and winning animals. Also, the reward scheme was designed by children in the participatory design group	Met
15	The system shall not use timed learning tasks	Timed tasks have not been used	Met

P.2 Non-functional requirements

Req.	Description	Analysis	Status
16	The system shall be able to be used on an interactive whiteboard	The evaluations using interactive whiteboards have shown that the system can be used on interactive whiteboards (see Section 6.3.3),	Met
17	The system shall be able to be used on a PC	The system functions normally on a PC	Met
18	The system shall be able to function without a hardware keyboard	The system functions without a hardware keyboard because an on-screen keyboard has been implemented and this can be used instead	Met
19	The system shall function on hardware used in Moorlands Junior School	The system functions on hardware used in Moorlands Junior School	Met

20	The system shall function on the operating system used in Moorlands Junior School	The system functions on the operating system used in Moorlands Junior School	Met
----	---	--	-----

P.3 User requirements

Req.	Description	Analysis	Status
21	The system shall have content for different years in key stage two	This has not been implemented, only year 5 has been covered due to high level design decisions (see Section 5.2), however, the infrastructure has been put in place for this to be extended in a full implementation	Partial
22	The system shall have content appropriate for different mathematical ability levels within the different years in key stage two	This has not been implemented, only middle ability has been covered due to high level design decisions (see Section 5.2), however, the infrastructure has been put in place for this to be extended in a full implementation	Partial
23	The system should provide tuition linked to familiar concepts	Animals are thought to be familiar to children as the theme was designed by children	Met
24	The system should provide tuition linked to concrete concepts	All questions make reference to animals and zoo related objects. The questions are made concrete in the help activities by pictorial representations of animals and other zoo related objects	Met
25	The system should use vocabulary familiar to the user	The formative evaluations identified vocabulary the children in year 5 had difficulty with, which resulted in changes being made to the vocabulary used such that the system should now use vocabulary familiar to children in year 5	Met

P.4 Usability requirements

Req.	Description	Analysis	Status
26	The system shall have a simple interface	This is difficult to judge without an expert evaluation but it is thought that the system has a simple interface as the children in the evaluations were able to use the system quite easily (see Section 6.2, 6.3 and 8.3.2 - 8.3.4)	Met
27	The system shall primarily use direct manipulation	Most interaction is by direct manipulation	Met
28	The system should make use of menu selection where appropriate	A menu has been implemented in the form of the 'menu bar', this is used by direct manipulation	Met
29	The system should make judicious use of form fillin	Form fillin is only used where the players enters their details and on the male lion help activity. It is possible to use an on-screen keyboard for this, thus allowing the input to be by direct manipulation	Met

30	The system shall use tool tips or labels on buttons	All menu buttons use labels as well as icons. All other buttons have labels, apart from where the user chooses their zoo keeper, jeep or animal for the certificate, but here the picture of the object is the button. Tool tips have been added to buttons that do not have labels e.g. the certificate button which appears over the zone when the certificate has been won.	Met
31	The system shall use high contrast colour combinations	All text buttons have a white background and black writing. Questions and advice are given in speech bubbles with black text on a white background. The game background is a pale yellow in most parts, with foreground objects using darker colours. There may not be enough contrast on the map with the text over the zones, because the text is black and the zones vary in colours. However, the children wanted a range of colours on the map.	Partial
32	The system should not use colours that people with colour blindness may have difficulty in distinguishing to indicate meaning	Colour is not used to distinguish between a right and wrong answer. This is made explicit by the text and pictures used on the question right page and question wrong page. Red/Green is the most common type of colour blindness and these colours are not used together anywhere	Met
33	The system should use large sans serif fonts that can be read from a reasonable distance from the display screen	Verdana, a sans serif font, has been used across the system and it thought to be a suitable size as the children were able to read the text when at a PC and when using the game on an interactive whiteboard (see Section 6.3.3)	Met
34	The system should not require the user to interact with the top half of the interface	The menu is located at the bottom of the screen and most other buttons are in the bottom half of the screen. It is possible to interact with the game by only using the bottom half of the interface. However, with the help activities due to the amount of space required to fit all the objects on the interface, it may be necessary for the user to interact just above the half way point	Partial
35	The system shall not require the user to 'write' with the pointing device	The user does not need to write with the pointing device	Met
36	The system should minimise typed input	Typed input is only required in three fields and this can be done using a mouse or pointing device with the on-screen keyboard	Met
37	The system shall not use verbal audio output	The system does not use verbal audio output	Met

38	The system should only display functionality necessary to complete a task	The menu only shows buttons that are appropriate to where the user is in the game i.e. the Print button is only visible when viewing the certificate	Met
39	The system should meet Shneiderman's (1998) 'Eight Golden Rules of Interface Design' (See Appendix A)	<ul style="list-style-type: none"> • <i>Strive for consistency</i> - This is difficult to judge without an expert evaluation, but it is thought that the system has used consistent sets of actions, terminology, colour, layout, fonts etc. • <i>Enable frequent users to use shortcuts</i> - A return shortcut can be used on the lion help activity, but no other keyboard shortcuts have been implemented • <i>Offer informative feedback</i> - As direct manipulation is used, feedback is visual. Feedback and advice is also given when questions are answered incorrectly • <i>Design dialogue to yield closure</i> - The process of completing a question can be considered a sequence of actions with a beginning, middle and end as can the process of completing a zone. For each feedback and reward is given. • <i>Offer error prevention and simple error handling</i> - The user is unable to make any serious errors. Error handling is used in the lion help activity to ensure only numbers are entered and feedback is given • <i>Permit easy reversal of actions</i> - When starting a new game, the player can go back and change their choices. The player is able to leave a question and help activity at any points without any negative effects • <i>Support internal locus of control</i> - The menu bar and map provides freedom in navigation, thus allowing the user to choose where they want to go and what they want to do • <i>Reduce short-term memory load</i> - The interface is simple and uses recognition rather than recall as all options are visible for the appropriate part of the game 	Partial

Q *Learning Evaluation Written Division Tests*

Based and adapted from questions in the 10Ticks (<http://www.10ticks.co.uk/>) Level 4 Pack 2.

Division Test 1

1. A milk crate will hold 24 bottles of milk. There are 4 rows of bottles in a milk crate.
How many bottles are in each row?
2. Hamish sells tulips in bunches of 8. He has 64 tulips. How many bunches can he make?
3. Angus sells roses in bunches of 6. He has 144 roses. How many bunches can he make?
4. Jenny, Bob and Carol win £42 between them on the pools. How much do they each get?
5. Jenny, Bob and Carol win £771 between them on the lottery. How much do they each get?
6. It is 63 days until Ben's birthday. How many weeks away is it?
7. It is 161 days until Tom's birthday. How many weeks away is it?
8. Fiona buys sherbet straws in the newsagents. They cost 4p each and she spends 28p.
How many straws has she bought?
9. Richard buys 54 Gob Stoppers. They are shared out between 6 of them. How many Gob Stoppers do they each get?
10. Jenny works out that her little baby brother, Herman, is 48 months old. How many years old is Herman?

Division Test 2

1. A milk crate will hold 24 bottles of milk. There are 6 rows of bottles in a milk crate.
How many bottles are in each row?
2. Hamish sells tulips in bunches of 8. He has 72 tulips. How many bunches can he make?
3. Angus sells roses in bunches of 6. He has 150 roses. How many bunches can he make?
4. Jenny, Bob and Carol win £36 between them on the pools. How much do they each get?
5. Jenny, Bob and Carol win £813 between them on the lottery. How much do they each get?
6. It is 56 days until Ben's birthday. How many weeks away is it?
7. It is 182 days until Tom's birthday. How many weeks away is it?
8. Fiona buys sherbet straws in the newsagents. They cost 4p each and she spends 32p.
How many straws has she bought?
9. Richard buys 48 Gob Stoppers. They are shared out between 6 of them. How many Gob Stoppers do they each get?
10. Jenny works out that her little baby brother, Herman, is 60 months old. How many years old is Herman?

R *Learning Evaluation Scripts*

Group 1 and Group 2

My name is Katherine and I am from the University of Bath. I have been working with some children from your school to design a computer game for maths. The game is called 'Maths Zoo' and at the moment, it focuses on the topic of division. Today I am going to carry out a study with your class to try and find out if the game helps you learn and understand division.

Group 1

Firstly, I would like each of you to do a division test with 10 questions in 5 minutes. From this I should be able to find out what you already know about division.

Then I will put you into pairs and you will have 25 minutes to play a new Maths Zoo game and try to complete the Super Safari Zone. If you complete the Super Safari Zone before the time is up, you should write down your score on the worksheet I give you, otherwise, you should write down your score at the end of the 25 minutes on this worksheet. From these scores, I will be able to see how well each pair has done when using the game.

The final task will be another division test, again with 10 questions and 5 minutes to do the test.

Group 2

The class will be split into two groups. If you are in the first group, I would like each of you to read about division from a sheet I give you, and then I would like you to complete a division test with 20 questions, so I can find out your current understanding of division. You will have 15 minutes to do the reading and the test. After this you will be put into pairs and you will have 20 minutes to play a new Maths Zoo game and try to complete the Super Safari Zone.

The other group will be put into pairs straight away and will have 20 minutes to play a new Maths Zoo game and try to complete the Super Safari Zone. I would then like each of you to read about division and do a division test with 20 questions, so I can find out your understanding of division after playing the game. You will have 15 minutes to do the reading and the test.

When you are playing the game if you complete the Super Safari Zone before the time is up, you should write down your score on the worksheet I give you, otherwise, you should write down your score at the end of the 20 minutes on the worksheet. From these scores, I will be able to see how well each pair has done when using the game.

Group 1 and Group 2

From the tests and the game scores, I will be able to make comparisons to see if there have been any improvements in your understanding of division from using the game. The findings from this study will go into my project report; however, I will not use any of your names in the report and the data will be kept confidential.

Thanks for helping me with my project.

S Learning Evaluation Results: Class 1

Table S.1 Results from Class 1 Learning Evaluation

Code	Partner Code	Test 1 score (from 10)	Test 2 score (from 10)	Test 1 and Test 2 score difference	Maths Zoo score (from 80)	Maths Zoo score standardized (from 10)
L1	M6	2	3	1	25	3.1
L2	L4, L3	1	1	0	50	6.3
L3	L2, L4	1	0	-1	50	6.3
L4	L2, L3	1	1	0	50	6.3
L5	L6	0	2	2	25	3.1
L6	L5	0	0	0	25	3.1
M1	M2	4	3	-1	40	5.0
M10	M5	1	2	1	50	6.3
M11	M8	0	2	2	40	5.0
M2	M1	4	1	-3	40	5.0
M3	H2	3	1	-2	45	5.6
M4	M7	3	0	-3	50	6.3
M5	M10	3	2	-1	50	6.3
M6	L1	2	2	0	25	3.1
M7	M4	1	2	1	50	6.3
M8	M11	1	2	1	40	5.0
M9	H3	1	4	3	80	10.0
H1	H4	8	8	0	80	10.0
H2	M3	8	3	-5	45	5.6
H3	M9	8	9	1	80	10.0
H4	H1	7	4	-3	80	10.0
H5	H6	6	4	-2	70	8.8
H6	H5	4	4	0	70	8.8

S.1 The Wilcoxon Signed Ranks Test applied to Class 1 Results

The Wilcoxon signed ranks test uses information on direction and relative magnitude within pairs of results (Siegel and Castellan 1988). This has been calculated as follows:

Null hypothesis H_0 : There is no difference between the Test 1 and Test 2 scores

Alternative hypothesis H_1 : There is a difference between the Test 1 and Test 2 scores

Significance level, $\alpha = 0.05$

Let X be Test 1 and Y be Test 2

Let d_i be the difference between the scores under X and Y, therefore: $d_i = X_i - Y_i$

Rank $|d_i|$ in ascending order, if $d_i = 0$ do not rank and where ranks are tied, give a mean rank

Give each rank a positive or negative value according to the direction of the difference (see table S.2)

Table S.2 Results from Class 1 Learning Evaluation manipulated for the Wilcoxon Signed Ranks Test

X	Y	d _i	d _i	Ranking
8	8	0	0	-
4	4	0	0	-
1	1	0	0	-
1	1	0	0	-
0	0	0	0	-
2	2	0	0	-
8	9	-1	1	-4.5
2	3	-1	1	-4.5
1	2	-1	1	-4.5
1	2	-1	1	-4.5
1	2	-1	1	-4.5
1	0	1	1	4.5
4	3	1	1	4.5
3	2	1	1	4.5
0	2	-2	2	-10.5
0	2	-2	2	-10.5
6	4	2	2	10.5
3	1	2	2	10.5
1	4	-3	3	-14.5
7	4	3	3	14.5
4	1	3	3	14.5
3	0	3	3	14.5
8	3	5	5	17

Let T^+ be the sum of ranks of positive d_i 's, thus $T^+ = 95$

Let N be the number of nonzero d_i 's, thus $N = 17$.

As $N > 15$, compute z as follows (Siegel and Castellan 1988, p.91.):

$$\begin{aligned}
 z &= \frac{T^+ - N(N+1)/4}{\sqrt{N(N+1)(2N+1)/24}} \\
 &= \frac{95 - (17)(18)/4}{\sqrt{(17)(18)(35)/24}} \\
 &= 0.86
 \end{aligned}$$

Determine the associated probability ρ , under H_0 of z (Siegel and Castellan 1988, p.319)

$\rho = 0.1949$, but as it is a two tailed test, use 2ρ , $2\rho = 0.3898$

If $2\rho \leq \alpha$, reject H_0

As $0.3898 > 0.05$, cannot reject H_0

H_0 is upheld: There is no difference between the Test 1 and Test 2 scores.

T Learning Evaluation Results: Class 2

T.1 Group A results (written test then system)

Table T.1 Results from Class 2 Groups A Learning Evaluation

Code	Partner code	Test score (from 20)	Maths Zoo score (from 80)	Maths Zoo standardized score (from 20)
AL1	AL2	3	20	5
AL2	AL1	2	20	5
AL3	AL4	1	15	3.8
AL4	AL3	0	15	3.8
AM1	AM4	10	15	3.8
AM2	AM3	9	40	10
AM3	AM2	7	40	10
AM4	AM1	5	15	3.8
AH1	AH2	13	75	18.8
AH2	AH1	12	75	18.8
AH3	AH4	6	60	15
AH4	AH3	6	60	15

T.2 Group B results (system then written test)

Table T.2 Results from Class 2 Groups B Learning Evaluation

Code	Partner code	Test score (from 20)	Maths Zoo score (from 80)	Maths Zoo standardized score (from 20)
BL1	BL4	3	20	5
BL2	BL3	3	80	20
BL3	BL2	3	80	20
BL4	BL1	3	20	5
BM1	BM2	2	60	15
BM2	BM1	1	60	15
BM3	BM4	0	30	7.5
BM4	BM3	0	30	7.5
BH1	BH2	13	80	20
BH2	BH1	11	80	20
BH3	BH4	10	80	20
BH4	BH3	9	80	20
BH5	BH6	7	70	17.5
BH6	BH5	6	70	17.5

T.3 The Robust Rank-Order Test Applied to the Test 1 Results

The Robust Rank-Order Test can be used to test a null hypothesis where there are two groups that are assumed to be independent and where it cannot be assumed that their underlying distributions are the same (Siegel and Castellan 1988). This has been calculated as follows:

Null hypothesis H_0 : There is no difference between the Group A and Group B Test scores

Alternative hypothesis H_1 : There is a difference between the Group A and Group B Test scores

Significance level, $\alpha = 0.05$

Let m be the number of children in group A, thus m = 12

Let n be the number of children in group B, thus n = 14

Combine and order the scores from both groups in ascending order

Give the scores in group A a placement, $U(BA_i)$, with adjustments made for ties

Give the scores in group B a placement $U(XY_j)$, with adjustments made for ties (see Table T.3).

Table T.3 Class 1 Test Results Ordered and Scored for the Robust Rank-Order Test

Group	Test Score	$U(BA_i)$	$(U(BA_i) - U(BA))^2$	$U(AB_j)$	$(U(AB_j) - U(AB))^2$
A	0.0	0.5	55.63		
B	0.0			0.5	21.56
B	0.0			0.5	21.56
A	1.0	2.5	29.79		
B	1.0			1.5	13.27
A	2.0	3.5	19.88		
B	2.0			2.5	6.98
A	3.0	6.0	3.84		
B	3.0			3.5	2.70
B	3.0			3.5	2.70
B	3.0			3.5	2.70
B	3.0			3.5	2.70
A	5.0	8.0	0.00		
A	6.0	8.5	0.29		
A	6.0	8.5	0.29		
B	6.0			6.0	0.73
A	7.0	9.5	2.38		
B	7.0			7.5	5.56
A	9.0	10.5	6.46		
B	9.0			8.5	11.27
A	10.0	11.5	12.54		
B	10.0			9.5	18.98
B	11.0			10.0	23.59
A	12.0	13.0	25.42		
A	13.0	13.5	30.71		
B	13.0			11.5	40.41
		Total	95.50	Total	72.00

Let $U(BA)$ and $U(AB)$ be the mean placements for Group A and B respectively (see Table T.4). Calculate these using the following equations:

$$U(BA) = \sum_{i=1}^m \frac{U(BA_i)}{m}$$

$$U(AB) = \sum_{j=1}^n \frac{U(AB_j)}{n}$$

Let V_a and V_b be the variability indices for groups A and B respectively (See Table T.4). Calculate these using the following equations:

$$V_a = \sum_{i=1}^m [U(BA_i) - U(BA)]^2$$

$$V_b = \sum_{j=1}^n [U(AB_j) - U(AB)]^2$$

Table T.4 Class 1 Test Results Mean placements and variability indices

U(BA)	7.96
U(AB)	5.14
V_a	187.23
V_b	174.71

Calculate the test statistic U using as follows:

$$\dot{U} = \frac{mU(BA) - nU(AB)}{2\sqrt{V_a + V_b + U(AB)U(BA)}}$$

$$= \frac{12(7.96) - 14(5.14)}{2\sqrt{187.23 + 174.71 + (5.14)(7.96)}}$$

$$= 0.58$$

Determine the associated probability p , of \dot{U} (Siegel and Castellan 1988, p.319).

$p = 0.2810$, as it is a two tailed test use $2p$, $2p = 0.5620$

If $2p \leq \alpha$, reject H_0

As $0.5620 > 0.05$, cannot reject H_0

H_0 is upheld: There is no difference between the Group A and Group B Test scores.

U User Experience Questionnaire Results

Pair	Maths Zoo Score	Safari Zone Complete?	What did you like about the Maths Zoo?	What didn't you like about the Maths Zoo?
L2, L3, L4	50	N	It was very fun	The questions were hard
L5, L6	25	N	You won animals	
AL1, AL2	20	N	Questions	Too much
AL3, AL4	15	Y	The games and the fun	It was the adding
BL2, BL3	80	Y	I liked winning the lion	
BL1,BL4	20	N	It was fun and we like it, it's groovy and we learnt things	
M1, M2	40	Y	Good questions	Counting
M4, M7	50	Y	It was fun and easy	It was too much fun
M6, L1	25	Y	That is was easy	The instructions were not clear
M8, M11	40	N	The animals	Hard division
AM1, AM4	15	Y	We like the bit where you can choose your own jeep and zoo keepers	The questions were hard
AM2, AM3	40	Y	It wasn't boring. You got another go if you got it wrong and you could have help	Division
BM1, BM2	60	Y	It helps us with our learning	Nothing
BM3, BM3	30	N	It was fun	
H1, H1	80	Y	Fun maths	Too easy questions
H2, M3	45	Y	It was fun	
H3, M9	80	Y	The way they used the zoo to create maths problems	Nothing
H5, H6	70	Y	It made you want to try your hardest because you wanted to get the animal	You get points deducted if you do it wrong
AH1, AH2	75	Y	It helps you with your division and it's fun	
AH3, AH4	60	Y	The animations	The lion because it was very scary
BH1, BH2	80	Y	Working out the answers, winning the animals	The help was a bit difficult to use
BH3, BH4	80	Y	It is fun and colourful and the maths were good	Nothing
BH5, BH6	70	Y	It's a fun way to learn	It was a bit too easy

V Source Code

The source code for the Maths Zoo 1.0 prototype is listed in this Appendix. The code is listed in terms of the layer and keyframe the code is found in. Descriptions of these layers and keyframes can be found in sections 7.4.2 and 7.4.3 respectively. The classes have not been listed again as these are detailed in section 7.4.1.

Layer	Actions	Keyframe	Splash
-------	---------	----------	--------

```

stop();
import zoo.*;
currentPage = "Splash";

//Hide menu function
function hideMenu():Void {
    menu._visible = false;
}

//Show menu function
function showMenu():Void {
    menu._visible = true;
}

//Hide the menu menu initially
hideMenu();

//Flash context menu visibility
Stage.showMenu = false;

// Navigation buttons action
newGame_btn.onRelease = function() {
    gotoAndStop("NewGame");
};
savedGame_btn.onRelease = function() {
    gotoAndStop("SavedGame");
};
instructions_btn.onRelease = function() {
    gotoAndStop("Instructions");
};
exit_btn.onRelease = function() {
    //Close the application
    fscommand("Quit", "");
};

//Drag and drop function
function dragAndDrop(clip) {
    clip.onPress = function() {
        startDrag(this);
    };
    clip.onRelease = function() {
        stopDrag();
    };
}

//Function to bring up and use the onscreen keyboard
function screenKeyboardEntry(fieldText:String, fieldName:String) {

```

```

//Attach and position the keyboard
_root.attachMovie("OnscreenKeyboard", "keyboard", 105);
keyboard._x = 95;
keyboard._y = 150;

//Check if any text has been entered using the keyboard
peripheral,
// if so show this on the onscreen keyboard
if (fieldText == undefined) {
    _root.textEntered = "";
} else {
    _root.textEntered = fieldText;
}

//by default have caps lock off
capitals = false;
_root.capitalsVal = "Capitals off";

//Turn caps lock on and off
keyboard.Capitals_btn.onRelease = function() {
    if (capitals == false) {
        capitals = true;
        _root.capitalsVal = "Capitals on";
    } else {
        capitals = false;
        _root.capitalsVal = "Capitals off";
    }
};

//Enter space when space button pressed
keyboard.Space_btn.onRelease = function() {
    _root.textEntered += " ";
};

//Delete the last character entered
keyboard.Delete_btn.onRelease = function() {
    if (_root.textEntered == undefined) {
        //No text has been entered yet, so don't need
to do anything
    } else {
        //Need to remove the last character entered
        _root.textEntered =
keyboard.textEntered.text.substr(0, keyboard.textEntered.text.length-
1);
    }
};

//Function to get letter from key press on onscreen keyboard

```

```

function keyboardValue(letter, normalLetter:String,
capitalLetter:String) {
    letter.onRelease = function() {
        if (capitals == false) {
            _root.textEntered += normalLetter;
        } else {
            _root.textEntered += capitalLetter;
        }
    };
}

//Cancel button actions, remove the keyboard from the screen
keyboard.cancelKeyboard_btn.onRelease = function() {
    //Close onscreen keyboard
    removeMovieClip(keyboard);
};

//Enter button actions, use the text entered
keyboard.enterKeyboard_btn.onRelease = function() {
    removeMovieClip(keyboard);
    if (fieldName == "playerName") {
        _root.playerName = _root.textEntered;
    } else if (fieldName == "gameName") {
        _root.gameName = _root.textEntered;
    } else if (fieldName == "hutMoney") {
        _root.hutMoney = _root.textEntered;
    }
};

//Keyboard value for each letter of the alphabet
keyboardValue(keyboard.A, "a", "A");
keyboardValue(keyboard.B, "b", "B");
keyboardValue(keyboard.C, "c", "C");
keyboardValue(keyboard.D, "d", "D");
keyboardValue(keyboard.E, "e", "E");
keyboardValue(keyboard.F, "f", "F");
keyboardValue(keyboard.G, "g", "G");
keyboardValue(keyboard.H, "h", "H");
keyboardValue(keyboard.I, "i", "I");
keyboardValue(keyboard.J, "j", "J");
keyboardValue(keyboard.K, "k", "K");
keyboardValue(keyboard.L, "l", "L");
keyboardValue(keyboard.M, "m", "M");
keyboardValue(keyboard.N, "n", "N");
keyboardValue(keyboard.O, "o", "O");
keyboardValue(keyboard.P, "p", "P");
keyboardValue(keyboard.Q, "q", "Q");
keyboardValue(keyboard.R, "r", "R");

```

```

keyboardValue(keyboard.S, "s", "S");
keyboardValue(keyboard.T, "t", "T");
keyboardValue(keyboard.U, "u", "U");
keyboardValue(keyboard.V, "v", "V");
keyboardValue(keyboard.W, "w", "W");
keyboardValue(keyboard.X, "x", "X");
keyboardValue(keyboard.Y, "y", "Y");
keyboardValue(keyboard.Z, "z", "Z");

```

```

//Keyboard value for each number
keyboardValue(keyboard.no1, "1", "1");
keyboardValue(keyboard.no2, "2", "2");
keyboardValue(keyboard.no3, "3", "3");
keyboardValue(keyboard.no4, "4", "4");
keyboardValue(keyboard.no5, "5", "5");
keyboardValue(keyboard.no6, "6", "6");
keyboardValue(keyboard.no7, "7", "7");
keyboardValue(keyboard.no8, "8", "8");
keyboardValue(keyboard.no9, "9", "9");
keyboardValue(keyboard.no0, "0", "0");

```

```

}

```

Layer	Menu	Keyframe	Splash
-------	------	----------	--------

```

//Function to remove any clips generated in the help activites
function removeClips():Void {
    if (currentPage == "ElephantActivity") {
        //Remove the food buckets that were added for the
        elephant help activity
        removeFoodClips();
        removeElephantClips();
    } else if (currentPage == "GiraffeActivity") {
        removeGiraffeClips();
        removeFieldClips();
    } else if (currentPage == "ZebraActivity") {
        removeHayClips();
        removeZebraClips();
    } else if (currentPage == "LionActivity2") {
        removeAnimalClips();
    }
}

```

```

//Save game function
function save():Void {

    //Attach the save game pop up and position
    _root.attachMovie("SaveGameWindow", "saveGamePopUp", 500);
    saveGamePopUp._x = 95;
}

```

```

saveGamePopUp._y = 150;

//Yes button actions, the player wishes to save the game
saveGamePopUp.yesSaveGame_btn.onRelease = function() {
    //Remove the first pop up and attach the game saved
notification pop up
    removeMovieClip(saveGamePopUp);
    _root.attachMovie("SavedGameWindow", "gameSavedPopUp",
501);

    gameSavedPopUp._x = 95;
    gameSavedPopUp._y = 150;

    //Player needs to acknowledge save by clicking OK
    gameSavedPopUp.OK_btn.onRelease = function() {
        //Check if the save has been called because the
player wants to exit
        if (exitGame == true) {
            gameSaved = true;
            removeMovieClip(gameSavedPopUp);
            //Exit the game
            fscommand("Quit", "");
        } else {
            //The player wants to carry on playing
after saving
            gameSaved = true;
            removeMovieClip(gameSavedPopUp);
        }
    };
};

//Yes button actions, the player wants to cancel the save
saveGamePopUp.noSaveGame_btn.onRelease = function() {
    removeMovieClip(saveGamePopUp);
    //Check if the player wants to exit
    if (exitGame == true) {
        fscommand("Quit", "");
    }
};
};

//Exit function
function exit():Void {
    //Attach the exit pop up and position
    _root.attachMovie("ExitWindow", "exitPopUp", 502);
    exitPopUp._x = 95;
    exitPopUp._y = 150;

    //Yes button actions, the player wants to exit the game

```

```

exitPopUp.yesExit_btn.onRelease = function() {
    exitGame = true;
    removeMovieClip(exitPopUp);
    //Check if they want to save the game before exiting
    if (gameSaved == false) {
        save();
    } else {
        removeMovieClip(exitPopUp);
        fscommand("Quit", "");
    }
};

//No button actions, the player wants to cancel the exit
exitPopUp.noExit_btn.onRelease = function() {
    exitGame = false;
    removeMovieClip(exitPopUp);
};
};

//Function to print the certificate
function printCertificate():Void {
    _root.attachMovie("PrintWindow", "printPopUp", 103);
    printPopUp._x = 95;
    printPopUp._y = 150;
    //Print button actions
    printPopUp.yesPrint_btn.onRelease = function() {
        print(finalCertificate, "bframe");
        removeMovieClip(printPopUp);
    };
    printPopUp.cancelPrint_btn.onRelease = function() {
        removeMovieClip(printPopUp);
    };
};

//Instruction button actions
menu.instructions_btn.onRelease = function() {
    gotoAndStop("Instructions");
};

//Question button actions
menu.question_btn.onRelease = function() {
    removeClips();
    gotoAndStop("Question");
};

//Map button actions
menu.map_btn.onRelease = function() {
    removeClips();

```

```

        gotoAndStop("Map");
    };

    //Zone button actions
    menu.zone_btn.onRelease = function() {
        removeClips();
        gotoAndStop("SafariZone");
    };

    //Certificate button actions
    menu.certificate_btn.onRelease = function() {
        gotoAndStop("Certificate");
    };

    //Print button actions
    menu.print_btn.onRelease = function() {
        printCertificate();
    };

    //Save button actions
    menu.save_btn.onRelease = function() {
        save();
    };

    //Exit button actions
    menu.exit_btn.onRelease = function() {
        exit();
    };

```

Layer	GameData	Keyframe	Splash
import zoo.*;			
/***** QUESTION Details *****/			
//Type of questions			
year = 5;			
difficulty = "Medium";			
/***** GIRAFFE QUESTIONS *****/			
//Giraffe Question 1			
//Question values and text that will be displayed on the question and activity page			
V1 = 12;			
V2 = 3;			
questionText = "Question:\nThere are "+ V1 +" giraffes but only "+ V2			
+ " giraffe fields. The zoo keeper wants to put			
"			
+ "equal numbers of giraffes in each field.			
How many "			

```

        + "giraffes should go in each field?";
    //Multiple choice answers
    A = 2;
    B = 3;
    C = 4;
    D = 5;
    //The correct answer
    correct = C;
    //Help text if the question is answered incorrectly
    help = "Remember there needs to be the same number of giraffes in each field.";
    //The help activity page for the question
    helpActivity = "GiraffeActivity";
    //Giraffe Question 1 object
    var giraffeQues1:Question = new Question(questionText, V1, V2, A, B, C, D, correct, year, difficulty, help, helpActivity);

    //Giraffe Question 2
    V1 = 24;
    V2 = 4;
    questionText = "Question:\nThere are " + V1 + " giraffes but only " + V2
    + " giraffe fields. The zoo keeper wants to
    put "
    + "equal numbers of giraffes in each field.
    How many "
    + "giraffes should go in each field?";
    A = 3;
    B = 6;
    C = 4;
    D = 5;
    correct = B;
    //Giraffe Question 2 object
    var giraffeQues2:Question = new Question(questionText, V1, V2, A, B, C, D, correct, year, difficulty, help, helpActivity);

    //Giraffe Question 3
    V1 = 28;
    V2 = 9;
    questionText = "Question:\nIf there are " + V1 + " giraffes and only " + V2
    + " giraffes can go in a field, how many
    fields are needed? ";
    A = 2;
    B = 3;
    C = 4;
    D = 5;
    correct = C;

```

```

help = "Remember, an extra field may be needed if there is a remainder
of giraffes.";
//Giraffe Question 3 object
var giraffeQues3:Question = new Question(questionText, V1, V2, A, B,
C, D, correct, year, difficulty, help, helpActivity);

//Giraffe Question 4
V1 = 29;
V2 = 6;
questionText = "Question:\nIf there are " + V1 + " giraffes and only "
+ V2
+ " giraffes can go in a field, how many fields
are needed? ";
A = 5;
B = 3;
C = 6;
D = 4;
correct = A;
//Giraffe Question 4 object
var giraffeQues4:Question = new Question(questionText, V1, V2, A, B,
C, D, correct, year, difficulty, help, helpActivity);

/***** ZEBRA QUESTIONS *****/
//Zebra Question 1
V1 = 8;
V2 = 32;
questionText = "Question:\nThere are " + V1 + " zebra stables and the
zoo keeper has " + V2
+ " bundles of hay to share between them, how
many bundles of hay "
+ "should go to each zebra stable?";

A = 4;
B = 2;
C = 5;
D = 3;
correct = A;
help = "Remember there needs to be the same number of hay bundles for
each zebra stable.";
helpActivity = "ZebraActivity";
//Zebra Question 1 object
var zebraQues1:Question = new Question(questionText, V1, V2, A, B, C,
D, correct, year, difficulty, help, helpActivity);

//Zebra Question 2
V1 = 7;
V2 = 35;
questionText = "Question:\nThere are " + V1 + " zebra stables and the
zoo keeper has " + V2
+ " bundles of hay to share between them, how
many bundles of hay "
+ "should go to each zebra stable?";

A = 4;
B = 5;
C = 2;
D = 3;
correct = B;
//Zebra Question 2 object
var zebraQues2:Question = new Question(questionText, V1, V2, A, B, C,
D, correct, year, difficulty, help, helpActivity);

//Zebra Question 3 object
V1 = 6;
V2 = 40;
questionText = "Question:\nThere are " + V1 + " zebras and each zebra
needs 7 bundles "
+ "of hay a week. If there are " + V2 + "
bundles of hay, how many more "
+ "bundles of hay does the zoo keeper need to
get?";
A = 1;
B = 2;
C = 3;
D = 4;
correct = B;
help = "Remember, you first need to work out the number of hay bundles
needed, then find the "
+ "difference between this and the number of hay bundles the
zoo keeper already has.";
//Zebra Question 3 object
var zebraQues3:Question = new Question(questionText, V1, V2, A, B, C,
D, correct, year, difficulty, help, helpActivity);

//Zebra Question 4
V1 = 5;
V2 = 30;
questionText = "Question:\nThere are " + V1 + " zebras and each zebra
needs 7 bundles "
+ "of hay a week. If there are " + V2 + "
bundles of hay, how many more "
+ "bundles of hay does the zoo keeper need to
get?";
A = 2;
B = 3;
C = 4;
D = 5;
correct = D;

```



```

//Zebra Question 4 object
var zebraQues4:Question = new Question(questionText, V1, V2, A, B, C,
D, correct, year, difficulty, help, helpActivity);

/***** ELEPHANT QUESTIONS *****/
//Elephant Question 1
V1 = 35;
V2 = 5;
questionText = "Question:\nThe zoo keeper has " + V1 + " buckets of
food for " + V2
he give to "
+ " elephants, how many buckets of food should
+ "each elephant so that they have the same?";
A = 5;
B = 6;
C = 7;
D = 8;
correct = C;
help = "Remember the elephants need the same number of food buckets
each.";
helpActivity = "ElephantActivity";
//Elephant Question 1 object
var elephantQues1:Question = new Question(questionText, V1, V2, A, B,
C, D, correct, year, difficulty, help, helpActivity);

//Elephant Question 2
V1 = 24;
V2 = 8;
questionText = "Question:\nThe zoo keeper has " + V1 + " buckets of
food for " + V2
he give to "
+ " elephants, how many buckets of food should
+ "each elephant so that they have the same?";
A = 2;
B = 3;
C = 4;
D = 5;
correct = B;
//Question Question 2 object
var elephantQues2:Question = new Question(questionText, V1, V2, A, B,
C, D, correct, year, difficulty, help, helpActivity);

//Elephant Question 3
V1 = 40;
V2 = 7;
questionText = "Question:\nIf there are " + V1 + " buckets of food and
an elephant needs "
+ "to eat " + V2 + " buckets of food, how many
elephants will this feed?";
A = 4;
B = 5;
C = 6;
D = 7;
correct = B;
help = "Remember, an elephant needs all "+V2+" buckets of food a day,
not just part of this.";
//Elephant Question 3 object
var elephantQues3:Question = new Question(questionText, V1, V2, A, B,
C, D, correct, year, difficulty, help, helpActivity);

//Elephant Question 4
V1 = 30;
V2 = 9;
questionText = "Question:\nIf there are " + V1 + " buckets of food and
an elephant needs "
+ "to eat " + V2 + " buckets of food, how many
elephants will this feed?";
A = 5;
B = 4;
C = 3;
D = 2;
correct = C;
help = "Remember, an elephant needs all "+V2+" buckets of food a day,
not just part of this.";
//Elephant Question 4 object
var elephantQues4:Question = new Question(questionText, V1, V2, A, B,
C, D, correct, year, difficulty, help, helpActivity);

/***** LION QUESTIONS *****/
//Lion Question 1
V1 = 8760;
V2 = 10;
questionText = "Question:\nThere is £" + V1 + " available to build "+
V2
+ " new identical lion huts, what is the
budget for each lion hut?";
A = "£860";
B = "£870";
C = "£867";
D = "£876";
correct = D;
help = "Remember, to divide by 10, the decimal place moves left by
one.";
helpActivity = "LionActivity";
//Lion Question 1 object

```

```

var lionQues1:Question = new Question(questionText, V1, V2, A, B, C,
D, correct, year, difficulty, help, helpActivity);

//Lion Question      2
V1 = 560;
V2 = 10;
questionText = "Question:\nThere is £" + V1 + " available to build " +
V2
+ " new identical lion huts, what is the budget
for each lion hut?";
A = "£56";
B = "£57";
C = "£50";
D = "£60";
correct = A;
//Lion Question 2 object
var lionQues2:Question = new Question(questionText, V1, V2, A, B, C,
D, correct, year, difficulty, help, helpActivity);

//Lion Question      3
V1 = 16;
V2 = 8;
questionText = "Question:\nThe zoo keeper starts work at midday and
needs to visit " + V1
+ " animals in " + V2 + " hours. If the zoo
keeper spends the same amount of "
+ "time with each animal, how long does the zoo
keeper spend with each animal?";
A = "One hour";
B = "Half an hour";
C = "Two hours";
D = "Quarter of an hour";
correct = B;
help = "Remember, one hour divided by two is half an hour and one hour
divided by four is a quarter of an hour";
helpActivity = "LionActivity2";
//Lion Question 2 object
var lionQues3:Question = new Question(questionText, V1, V2, A, B, C,
D, correct, year, difficulty, help, helpActivity);

//Lion Question 4
V1 = 16;
V2 = 4;
questionText = "Question:\nThe zoo keeper starts work at midday and
needs to visit " + V1
+ " animals in " + V2 + " hours. If the zoo
keeper spends the same amount of "
+ "time with each animal, how long does the
zoo keeper spend with each animal?";
correct = D;
//Lion Question 4 object
var lionQues4:Question = new Question(questionText, V1, V2, A, B, C,
D, correct, year, difficulty, help, helpActivity);

//Question Arrays for each animal
var mGiraffeQues:Array = [giraffeQues1, giraffeQues2];
var fGiraffeQues:Array = [giraffeQues3, giraffeQues4];
var mZebraQues:Array = [zebraQues1, zebraQues2];
var fZebraQues:Array = [zebraQues3, zebraQues4];
var mElephantQues:Array = [elephantQues1, elephantQues2];
var fElephantQues:Array = [elephantQues3, elephantQues4];
var mLionQues:Array = [lionQues1, lionQues2];
var fLionQues:Array = [lionQues3, lionQues4];

/***** ANIMAL OBJECTS *****/
var mGiraffe:Animal = new Animal("giraffe", "male", false,
mGiraffeQues);
var fGiraffe:Animal = new Animal("giraffe", "female", false,
fGiraffeQues);
var mZebra:Animal = new Animal("zebra", "male", false, mZebraQues);
var fZebra:Animal = new Animal("zebra", "female", false, fZebraQues);
var mElephant:Animal = new Animal("elephant", "male", false,
mElephantQues);
var fElephant:Animal = new Animal("elephant", "female", false,
fElephantQues);
var mLion:Animal = new Animal("lion", "male", false, mLionQues);
var fLion:Animal = new Animal("lion", "female", false, fLionQues);

//Array of all safari animals
var safariAnimals:Array = [mGiraffe, fGiraffe, mZebra, fZebra,
mElephant, fElephant, mLion, fLion];

//Empty array for other animals
var otherAnimals:Array = null;

//Function to choose a random question to start with
function setRandomStartQuestion(animals):Void {
    animal._questionNo =
Math.floor(Math.random()*animal._questionStore.length);
}

//Choose a start question randomly for each animal
for (i=0; i<safariAnimals.length; i++) {
    setRandomStartQuestion(safariAnimals[i]);
}

```

```

/***** ZONE OBJECTS *****/
//Only safari Zone has questions, animals, a topic and objectives
safariObjectives = "<li>Practice divison by numbers up to
10</li>"+ "<li>Practice divison with word problems</li>"+ "<li>Practice
division of money</li>"+ "<li>Practice division using time</li>";
var safariZone:Zone = new Zone("Super Safari", safariAnimals,
"Division", safariObjectives);

//Other zones do not have questions, animals, topics and objectives
for prototype
var mammalsZone:Zone = new Zone("Mammals", otherAnimals, "?", "?");
var birdsZone:Zone = new Zone("Brilliant Birds", otherAnimals, "?",
"?");
var reptilesZone:Zone = new Zone("Reptiles", otherAnimals, "?", "?");
var creepyCrawliesZone:Zone = new Zone("Creepy Crawlies",
otherAnimals, "?", "?");
var seaworldZone:Zone = new Zone("Seaworld", otherAnimals, "?", "?");
var primatesZone:Zone = new Zone("Primates Park", otherAnimals, "?",
"?");
var pandasZone:Zone = new Zone("Panda Paradise", otherAnimals, "?",
"?");
var cafeZone:Zone = new Zone("Cafe", otherAnimals, "?", "?");

//Array of zones
var zones:Array = [safariZone, mammalsZone, birdsZone, reptilesZone,
creepyCrawliesZone, seaworldZone, primatesZone, pandasZone, cafeZone];

/***** GAME OBJECT *****/
var game:Game = new Game(0, "", zones);
//Game Score
_root.gamePoints = game._gamePoints;

```

Layer	Actions	Keyframe	NameGame
	stop(); import zoo.*; currentPage = "NewGame"; hideMenu();		
	//Save and exit status var gameSaved:Boolean = true; var exitGame:Boolean = false;		
	// Game option variables schoolYear = 0; difficulty = "";		
	//Initial instructions and error messages		

```

newGameInstructions = "Choose your school year and game difficulty:";
newGameErrors = "";

// No options have been selected
year5Selected._visible = false;
mediumSelected._visible = false;
playerName = "";
gameName = "";

//Option button actions
year5_btn.onRelease = function() {
    year5Selected._visible = true;
    schoolYear = 5;
};
medium_btn.onRelease = function() {
    mediumSelected._visible = true;
    difficulty = "Medium";
};

//Navigation button actions
backToSplash_btn.onRelease = function() {
    gotoAndStop("Splash");
};

nextToDetails_btn.onRelease = function() {
    //Check all required options have been selected before moving
    on
        if (schoolYear == 0 || difficulty == "") {
            //Remove the instruction message, replace with an error
            message
                newGameInstructions = "";
                newGameErrors = "You have not selected:\n\n";

                if(schoolYear == 0){
                    newGameErrors += "<li>Your school year</li>";
                }
                if(difficulty == "") {
                    newGameErrors += "<li>The game
difficulty</li>";
                }
            }
        else {
            //Update the game object with school year and
            difficulty
                game._schoolYear = schoolYear;
                game._difficulty = difficulty;
                gotoAndStop("PlayerDetails");
            }
        }
    }

```

Layer	Actions	Keyframe	PlayerDetails
	<pre> stop(); import zoo.*; currentPage = "PlayerDetails"; hideMenu(); //Initial instructions and error messages playerDetailInstructions = "Enter your name and a name for your game:"; playerDetailErrors = ""; //Bring up keyboard to enter player name on Interactive whiteboard playerNameKeyboard_btn.onRelease = function() { screenKeyboardEntry(_root.playerName, "playerName"); }; //Bring up keyboard to enter game name on Interactive whiteboard gameNameKeyboard_btn.onRelease = function() { screenKeyboardEntry(_root.gameName, "gameName"); }; //Navigation button actions backToJeep_btn.onRelease = function() { gotoAndStop("NewGame"); }; nextToKeeper_btn.onRelease = function() { //Check all necessary information has been entered before moving on if (playerName == "" gameName == "") { //Remove the instruction message, replace with an error message playerDetailInstructions = ""; playerDetailErrors = "You have not entered:\n\n"; if(playerName == ""){ playerDetailErrors += "Your name"; } if (gameName == "") { playerDetailErrors += "A name for your game"; } } else { //update the game object with the player name and game name </pre>		

Layer	Actions	Keyframe	ChooseKeeper
	<pre> stop(); currentPage = "ChoosePlayer"; hideMenu(); //No zoo keeper has been selected yet zooKeeper = 0; //Initial instructions chooseKeeperInstructions = "Choose your Zoo Keeper:"; //Function to deselect all zoo keepers function deselectKeepers():Void { Keeper1Selected._visible = false; Keeper2Selected._visible = false; Keeper3Selected._visible = false; Keeper4Selected._visible = false; } //Start with no zoo keepers selected deselectKeepers(); //Function to select a zoo keeper function selectKeeper(keeper) { deselectKeepers(); keeper._visible = true; } //Buttons to select a zoo keeper Keeper1Selected_btn.onRelease = function() { zooKeeper = 1; selectKeeper(Keeper1Selected); }; Keeper2Selected_btn.onRelease = function() { zooKeeper = 2; selectKeeper(Keeper2Selected); }; Keeper3Selected_btn.onRelease = function() { zooKeeper = 3; selectKeeper(Keeper3Selected); }; </pre>		

```

Keeper4Selected_btn.onRelease = function() {
    zooKeeper = 4;
    selectKeeper(Keeper4Selected);
};

//Navigation button actions
backToOptions_btn.onRelease = function() {
    gotoAndStop("PlayerDetails");
};

nextToJeep_btn.onRelease = function() {
    //Check a zoo keeper has been selected before moving on
    if (zooKeeper == 0) {
        chooseKeeperInstructions = "You have not selected your
Zoo Keeper!";
    }
    else {
        game._zooKeeper = zooKeeper;
        gotoAndStop("ChooseJeep");
    }
};

```

Layer	Actions	Keyframe	ChooseJeep
	stop();		
	currentPage = "ChooseJeep";		
	hideMenu();		
	//No jeep is initially selected		
	chosenJeep = "";		
	//Initial instructions		
	jeepInstructions = "Choose your jeep:";		
	//Function to deselect all jeeps		
	function deselectJeeps():Void {		
	greenSelected._visible = false;		
	redSelected._visible = false;		
	yellowSelected._visible = false;		
	}		
	//Start with no jeeps selected		
	deselectJeeps();		
	//function to select a jeep		
	function selectJeep(jeep):Void {		
	deselectJeeps();		
	jeep._visible = true;		
	}		

```

//Jeep button actions
greenJeep_btn.onRelease = function() {
    chosenJeep = "green";
    selectJeep(greenSelected);
};

redJeep_btn.onRelease = function() {
    chosenJeep = "red";
    selectJeep(redSelected);
};

yellowJeep_btn.onRelease = function() {
    chosenJeep = "yellow";
    selectJeep(yellowSelected);
};

//Navigation button actions
start_btn.onRelease = function() {
    //Check a jeep has been selected before moving on
    if (chosenJeep == "") {
        jeepInstructions = "You have not chosen a jeep!";
    } else {
        game._jeep = chosenJeep;
        gotoAndStop("Map");
    }
};

backToKeeper_btn.onRelease = function() {
    gotoAndStop("ChooseKeeper");
};

```

Layer	Actions	Keyframe	SavedGame
	stop();		
	currentPage = "SavedGame";		
	hideMenu();		
	//Navigation button actions		
	backToSplash_btn.onRelease = function() {		
	gotoAndStop("Splash");		
	}		
	newGame_btn.onRelease = function() {		
	gotoAndStop("NewGame");		
	}		

Layer	Actions	Keyframe	Map
-------	---------	----------	-----

```

stop();
currentPage = "Map";
showMenu();

//Menu item visibility
menu.instructions_btn._visible = true;
menu.help_btn._visible = false;
menu.question_btn._visible = false;
menu.map_btn._visible = false;
menu.zone_btn._visible = false;
menu.print_btn._visible = false;
menu.certificate_btn._visible = false;
menu.save_btn._visible = true;
menu.exit_btn._visible = true;

//No zone selected anymore
zone = null;

//Key values to access objects in the zone array
var safari:Number= 0;
var mammals:Number = 1;
var birds:Number = 2;
var reptiles:Number = 3;
var creepy:Number = 4;
var seaworld:Number = 5;
var primates:Number = 6;
var pandas:Number = 7;
var cafe:Number = 8;

//function to get string with zone progress and topic
function zoneProgress(zoneName):String{
    if (zoneName._zoneName == "Super Safari"){
        progressText = "Topic: " + zoneName._topic +
"\n\nProgress:\n"+zoneName._noAnimalsWon+"/ "+zoneName._noAnimals;
    }
    else{
        progressText = "Progress:\n0/0";
    }
    return progressText;
}

//Generate text to go on each zone button indicating progress
safariProgress = zoneProgress(game._zoneStore[safari]);
mammalsProgress = zoneProgress(game._zoneStore[mammals]);
birdsProgress = zoneProgress(game._zoneStore[birds]);
reptilesProgress = zoneProgress(game._zoneStore[reptiles]);
creepyProgress = zoneProgress(game._zoneStore[creepy]);

```

```

seaworldProgress = zoneProgress(game._zoneStore[seaworld]);
primatesProgress = zoneProgress(game._zoneStore[primates]);
pandasProgress = zoneProgress(game._zoneStore[pandas]);

//Function to generate actions for zone button
function toZone(zoneKey, button):Void{
    button.onRelease = function() {
        zoneIndex = zoneKey;
        selectedZone = game._zoneStore[zoneIndex];
        gotoAndPlay("Jeep");
    };
}

//Generate actions for each of the zone
toZone(safari, map.safari_btn);
toZone(mammals, map.mammals_btn);
toZone(birds, map.birds_btn);
toZone(reptiles, map.reptiles_btn);
toZone(seaworld, map.seaworld_btn);
toZone(creepy, map.crawlies_btn);
toZone(primates, map.primates_btn);
toZone(pandas, map.panda_btn);
toZone(cafe, map.cafe_btn);

//Zone certificate visibility
safariCertificate_btn._visible = false;

//Show certificate to safari certificate if all questions are correct
if (game._zoneStore[safari]._noAnimals ==
game._zoneStore[safari]._noAnimalsWon) {
    //Show the certificate button
    safariCertificate_btn._visible = true;
    safariCertificate_btn.onRelease = function() {
        zone = game._zoneStore[safari];
        gotoAndStop("Certificate");
    }
};

```

Layer	Actions	Keyframe	Jeep
-------	---------	----------	------

```

stop();
currentPage = "Jeep";
showMenu();

//Menu button visibility
menu.instructions_btn._visible = true;
menu.help_btn._visible = false;
menu.question_btn._visible = false;

```

```

menu.map_btn._visible = true;
menu.zone_btn._visible = false;
menu.print_btn._visible = false;
menu.certificate_btn._visible = false;
menu.save_btn._visible = true;
menu.exit_btn._visible = true;

//Description of zone
zoneDetails = "Zone: "+selectedZone._zoneName+"\nTopic: " +
selectedZone._topic+"\nObjectives: "+selectedZone._objectives;

//Function to show only the selected jeep jeep
function showSelectedJeep():Void {
    //Hide all jeeps initially
    jeepGreen._visible = false;
    jeepRed._visible = false;
    jeepYellow._visible = false;

    //Decide which jeep to show based on jeep chosen at start of
game
    if (game._jeep == "green") {
        jeepGreen._visible = true;
    }
    else if (game._jeep == "red") {
        jeepRed._visible = true;
    }
    else if (game._jeep == "yellow") {
        jeepYellow._visible = true;
    }
}

//Function to show only one road items
function showRoadItem():Void {
    //Hide al road items initially
    trafficLights._visible = false;
    cone._visible = false;
    animalCrossing._visible = false;
    zebraCrossing._visible = false;

    //Decide which road item to show
    if (zoneIndex == 0 || zoneIndex == 4 || zoneIndex == 8 ) {
        trafficLights._visible = true;
    } else if (zoneIndex == 1 || zoneIndex == 5) {
        cone._visible = true;
    } else if (zoneIndex == 2 || zoneIndex == 6) {
        animalCrossing._visible = true;
    } else if (zoneIndex == 3 || zoneIndex == 7) {
        zebraCrossing._visible = true;
    }
}

```

```

    }
}

//Show the appropriate jeep and road item
showSelectedJeep();
showRoadItem();

//Navigation button actions
nextToZone_btn.onRelease = function() {
    if (selectedZone._zoneName == "Super Safari") {
        gotoAndStop("SafariZone");
    } else {
        //The other zones have not been built yet, so have a
generic page
        gotoAndStop("OtherZones");
    }
};


```

Layer	Actions	Keyframe	SafariZone
	stop();		
	currentPage = "SafariZone";		
	showMenu();		
	//menu button visibility menu.instructions_btn._visible = true; menu.help_btn._visible = false; menu.question_btn._visible = false; menu.map_btn._visible = true; menu.zone_btn._visible = false; menu.print_btn._visible = false; menu.certificate_btn._visible = false; menu.save_btn._visible = true; menu.exit_btn._visible = true;		
	//Initial greeting safariGreeting = "Welcome to the " + selectedZone._zoneName + " zone";		
	//Function to show only the selected zoo keeper function showSelectedKeeper():Void { //Hide all zoo keepers initially keeper1._visible = false; keeper2._visible = false; keeper3._visible = false; keeper4._visible = false;		
	//Show only zoo keeper that was selected at the start of the game if (game._zooKeeper == 1) { keeper1._visible = true;		

```

    }
    else if (game._zooKeeper == 2) {
        keeper2._visible = true;
    }
    else if (game._zooKeeper == 3) {
        keeper3._visible = true;
    }
    else if (game._zooKeeper == 4) {
        keeper4._visible = true;
    }
}

//Only show the selected zoo keeper
showSelectedKeeper();

//Have all the questions been answered
if (game._zoneStore[safari]._noAnimals ==
game._zoneStore[safari]._noAnimalsWon) {
    //Show that the zone has been completed and the certificate is
    available
    zoneGreeting = game._zoneStore[safari]._zoneName + " zone
    completed";
    zoneInstructions = "You have completed the " +
    game._zoneStore[safari]._zoneName + " zone, don't forget to get your
    certificate!";
    menu.certificate_btn._visible = true;
}
else {
    zoneInstructions = "Try and win a male and\nfemale of each type
    of animal by answering maths questions";
}

//Function to find out the win status of an animal and show either the
animal movie clip if it has been won
//or the button to win the animal if it has not yet been won
function animalStatus(animalIndexNo, animalMovieClip,
animalButton):Void {
    //Has the male giraffe been won?
    if (selectedZone._animalStore[animalIndexNo]._won == false) {
        //Do not show the picture of the giraffe, show the
        button to get the giraffe
        animalMovieClip._visible = false;
        animalButton._visible = true;

        //button to get to question
        animalButton.onRelease = function() {
            selectedAnimal =
selectedZone._animalStore[animalIndexNo];

```

```

        animalIndex = animalIndexNo;
        gotoAndStop("Question");
    }
    else {
        //Show the giraffe, do not show the button to get the
        giraffe
        animalMovieClip._visible = true;
        animalButton._visible = false;
    }
}

//Keys to access animals in zone animal store
mGiraffe = 0;
fGiraffe = 1;
mZebra = 2;
fZebra = 3;
mElephant = 4;
fElephant = 5;
mLion = 6;
fLion = 7;

//Find out if animals have been won, if they have been won, show the
animals
//in the table cell otherwise show a button to win the animal in the
table cell
animalStatus(mGiraffe, maleGiraffe, getMgiraffe_btn);
animalStatus(fGiraffe, femaleGiraffe, getFgiraffe_btn);
animalStatus(mZebra, maleZebra, getMzebra_btn);
animalStatus(fZebra, femaleZebra, getFzebra_btn);
animalStatus(mElephant, maleElephant, getMelephant_btn);
animalStatus(fElephant, femaleElephant, getFelephant_btn);
animalStatus(mLion, maleLion, getMlion_btn);
animalStatus(fLion, femaleLion, getFlion_btn);

```

Layer	Actions	Keyframe	OtherZones
	stop();		
	currentPage = "OtherZones";		
	showMenu();		
	//Menu button visibility menu.instructions_btn._visible = true; menu.help_btn._visible = false; menu.question_btn._visible = false; menu.map_btn._visible = true; menu.zone_btn._visible = false;		


```

menu.print_btn._visible = false;
menu.certificate_btn._visible = false;
menu.save_btn._visible = true;
menu.exit_btn._visible = true;

//Zone information
zoneInformation = "The "+ selectedZone._zoneName+" Zone is under
construction, but the Safari Zone is ready, so you could go there.";

//Show the picture for the selected zone only
function showSelectedZone(selectedZone):Void {

    mammalsZonePicture._visible = false;
    birdsZonePicture._visible = false;
    reptilesZonePicture._visible = false;
    seaworldZonePicture._visible = false;
    creepyCrawliesZonePicture._visible = false;
    primatesZonePicture._visible = false;
    cafeZonePicture._visible = false;
    pandaZonePicture._visible = false;

    //Show relevant zone picture
    if (selectedZone._zoneName == "Mammals") {
        mammalsZonePicture._visible = true;
    } else if (selectedZone._zoneName == "Brilliant Birds") {
        birdsZonePicture._visible = true;
    } else if (selectedZone._zoneName == "Reptiles") {
        reptilesZonePicture._visible = true;
    } else if (selectedZone._zoneName == "Seaworld") {
        seaworldZonePicture._visible = true;
    } else if (selectedZone._zoneName == "Creepy Crawlies") {
        creepyCrawliesZonePicture._visible = true;
    } else if (selectedZone._zoneName == "Primates Park") {
        primatesZonePicture._visible = true;
    } else if (selectedZone._zoneName == "Cafe") {
        cafeZonePicture._visible = true;
    } else if (selectedZone._zoneName == "Panda Paradise") {
        pandaZonePicture._visible = true;
    }
}

showSelectedZone(selectedZone);

//Navigation button actions
safariZone_btn.onRelease = function() {
    selectedZone = game._zoneStore[safari];
    zoneIndex = safari;
    gotoAndStop("SafariZone");

```

```

};

```

Layer	Actions	Keyframe	Question
-------	---------	----------	----------

```

stop();
currentPage = "Question";
showMenu();

//menu button visibility
menu.instructions_btn._visible = false;
menu.help_btn._visible = true;
menu.question_btn._visible = false;
menu.map_btn._visible = true;
menu.zone_btn._visible = true;
menu.print_btn._visible = false;
menu.certificate_btn._visible = false;
menu.save_btn._visible = true;
menu.exit_btn._visible = true;

//Only show the selected zoo keeper
showSelectedKeeper();

//Function show only the selected animal
function showAnimal(selectedAnimal):Void {
    //Hide all animals initially
    giraffe._visible = false;
    zebra._visible = false;
    elephant._visible = false;
    lion._visible = false;

    //Only show the selected animal
    if (selectedAnimal._animal == "giraffe") {
        giraffe._visible = true;
    } else if (selectedAnimal._animal == "zebra") {
        zebra._visible = true;
    } else if (selectedAnimal._animal == "elephant") {
        elephant._visible = true;
    } else if (selectedAnimal._animal == "lion") {
        lion._visible = true;
    }
}

//Only show the selected animal
showAnimal(selectedAnimal);

//The question
question = selectedAnimal._questionStore[selectedAnimal._questionNo];
questionTextDisplay = question._questionText;

```

```

//The help activity
helpAct = question._activityPage;

//Multiple choice answer button values
answerA = question._answers[0];
answerB = question._answers[1];
answerC = question._answers[2];
answerD = question._answers[3];

//The right answer
rightAnswer = question._correctAnswer;

//Function to check if selected answer is right or wrong
function checkAnswer(answer, answerButton):Void{
    answerButton.onRelease = function() {
        if (answer == rightAnswer) {
            gotoAndStop("QuestionRight");
        }
        else {
            gotoAndStop("QuestionWrong");
        }
    }
}

//Check if right or wrong answer for each multiple choice answer
checkAnswer(answerA, answerA_btn);
checkAnswer(answerB, answerB_btn);
checkAnswer(answerC, answerC_btn);
checkAnswer(answerD, answerD_btn);

//Help button actions
menu.help_btn.onRelease = function() {
    gotoAndStop(helpAct);
};

stop();
currentPage = "GiraffeActivity";
showMenu();

//menu button visibility
menu.instructions_btn._visible = false;
menu.help_btn._visible = false;
menu.question_btn._visible = true;
menu.map_btn._visible = true;
menu.zone_btn._visible = true;
menu.print_btn._visible = false;
menu.certificate_btn._visible = false;

```

Layer	Actions	Keyframe	GiraffeActivity
-------	---------	----------	-----------------

```

menu.save_btn._visible = true;
menu.exit_btn._visible = true;

//Only show the selected zoo keeper
showSelectedKeeper();

if(selectedAnimal._sex == "male"){
    //Instructional text for male giraffe questions
    _root.activityHint = "Move the fields\nand giraffes so that
there "
                                + "are the same number of giraffes in
each field";
}
else{
    //Instructional text for male giraffe questions
    _root.activityHint = "Try and fill each field with " +
question._value2
                                + " giraffes to see how many fields
are needed";
}

//Generate the appropriate number of food buckets to match the
question
for (i = 0; i < question._value1; i++){
    //Dynamic name for clip
    var giraffeClip:String = "giraffe" + i;

    //Attach clip
    this.attachMovie("giraffeSymbol", giraffeClip, 300+i);

    //Set the size of the clip
    (eval(giraffeClip))._width = 60;
    (eval(giraffeClip))._height = 80;

    //Position the clips in rows of 10
    if (i < 10){
        (eval(giraffeClip))._x = 750 - (40*i);
        (eval(giraffeClip))._y = 420;
    }
    else if(i >= 10 && i < 20){
        (eval(giraffeClip))._x = 750 - (40*(i-10));
        (eval(giraffeClip))._y = 340;
    }
    else {
        (eval(giraffeClip))._x = 750 - (40*(i-20));
        (eval(giraffeClip))._y = 260;
    }
}
//Make the clips drag and drop

```

```

        dragAndDrop(eval(giraffeClip));
    }

    //Function to remove the clips when the activity is finished
    function removeGiraffeClips():Void{
        for (i = 0; i < question._value1; i++){
            var giraffeClip:String = "giraffe" + i;
            removeMovieClip(eval(giraffeClip));
        }
    }

    //Work out the number of fields that need to be added
    if (selectedAnimal._sex == "male"){
        //Just use the second value in the question
        var numberOfFields:Number = question._value2;
    }
    else{
        //use the maximum multiple choice answer
        var maxAnswerAB:Number = Math.max(Number(answerA),
        Number(answerB));
        var maxAnswerCD:Number = Math.max(Number(answerC),
        Number(answerD));
        var maxAnswer:Number = Math.max(maxAnswerAB, maxAnswerCD);
        var numberOfFields:Number = maxAnswer;
    }

    //Generate the appropriate number of fields for the question
    for (i = 0; i < numberOfFields; i++){
        //dynamic variable name
        var fieldClip:String = "field" + i;

        //attach the clips, size and position on the stage
        this.attachMovie("field", fieldClip, 200+i);
        (eval(fieldClip))._width = 150;
        (eval(fieldClip))._height = 230;
        (eval(fieldClip))._x = 150 + (50*(i-1));
        (eval(fieldClip))._y = 275;

        //Make the clips drag and drop
        dragAndDrop(eval(fieldClip));
    }

    //Function to remove the clips when the activity is finished
    function removeFieldClips():Void{
        for (i = 0; i < numberOfFields; i++){
            var fieldClip:String = "field" + i;
            removeMovieClip(eval(fieldClip));
        }
    }

```

```

    }

```

Layer	Actions	Keyframe	ZebraActivity
	stop(); currentPage = "ZebraActivity"; showMenu();		
	//menu button visibility menu.instructions_btn._visible = false; menu.help_btn._visible = false; menu.question_btn._visible = true; menu.map_btn._visible = true; menu.zone_btn._visible = true; menu.print_btn._visible = false; menu.certificate_btn._visible = false; menu.save_btn._visible = true; menu.exit_btn._visible = true;		
	//Only show the selected zoo keeper showSelectedKeeper();		
	if(selectedAnimal._sex == "male"){ //Instructions text for male zebra question _root.activityHint = "Move the zebras\nand hay so that each zebra has the " }		+ "same number of hay bundles";
	} else{ //Instructional text for female zebra question _root.activityHint = "Try to move 7\nbundles of hay to each zebra, " }		+ "to see how many more bundles are needed for every zebra to have " + "7 bundles of hay";
	} //Generate the appropriate number of hay bundles to match the question for (i = 0; i < question._value2; i++){ //Dynamic variable name var hayClip:String = "hay" + i; }		
	//Attach and re-size the clips this.attachMovie("haySymbol", hayClip, 300+i); (eval(hayClip))._width = 60; (eval(hayClip))._height = 40;		
	//Positions the clips in rows of 20 on the stage		

```

if (i < 20){
    (eval(hayClip))._x = 730 - (30*i);
    (eval(hayClip))._y = 420;
}
else if (i>=20 && i<40){
    (eval(hayClip))._x = 730 - (30*(i-20));
    (eval(hayClip))._y = 380;
}
else if (i>=40 ){
    (eval(hayClip))._x = 730 - (30*(i-40));
    (eval(hayClip))._y = 340;
}
//Make the clips drag and drop
dragAndDrop(eval(hayClip));
}

//Function to remove the clips when the activity is finished
function removeHayClips():Void{
    for (i = 0; i < question._value2; i++){
        var hayClip:String = "hay" + i;
        removeMovieClip(eval(hayClip));
    }
}

//Generate the appropriate number of zebra to match the question
for (i = 0; i < question._value1; i++){
    //dynamic variable name
    var zebraClip:String = "zebra" + i;

    //Attach, resize and position the clips on the stage
    this.attachMovie("ZebraSymbol", zebraClip, 200+i);
    (eval(zebraClip))._width = 90;
    (eval(zebraClip))._height = 90;
    (eval(zebraClip))._x = 100 + (90*(i-1));
    (eval(zebraClip))._y = 200;

    //Make the clips drag and drop
    dragAndDrop(eval(zebraClip));
}

//Function to remove the clips when the activity is finished
function removeZebraClips():Void{
    for (i = 0; i < question._value1; i++){
        var zebraClip:String = "zebra" + i;
        removeMovieClip(eval(zebraClip));
    }
}

```

Layer	Actions	Keyframe	ElephantActivity
	stop(); currentPage = "ElephantActivity"; showMenu();		
	//menu button visibility menu.instructions_btn._visible = false; menu.help_btn._visible = false; menu.question_btn._visible = true; menu.map_btn._visible = true; menu.zone_btn._visible = true; menu.print_btn._visible = false; menu.certificate_btn._visible = false; menu.save_btn._visible = true; menu.exit_btn._visible = true;		
	//Only show the selected zoo keeper showSelectedKeeper();		
	if(selectedAnimal._sex == "female"){ //Instructional text for female elephant question _root.activityHint = "Try to move\n" + question._value2 + " buckets of food to each elephant " + "to see how many elephants can be fed all " + question._value2 + " buckets of food"; } else{ //Instructional text for male elephant question _root.activityHint = "Move the\nnbuckets of food to\nthe elphants so that they each have the same number"; }		
	//Generate the appropriate number of food buckets to match the question for (i = 0; i < question._value1; i++){ //dynamic variable name var foodClip:String = "foodBuckets" + i; //Attach and resize the clips this.attachMovie("bucket", foodClip, 200+i); (eval(foodClip))._width = 30; (eval(foodClip))._height = 50; //Position the clips in rows of 20 on the stage if (i < 20){ (eval(foodClip))._x = 750 - (20*i); (eval(foodClip))._y = 420;		

```

    }
    else if (i >= 20 && i < 40){
        (eval(foodClip))._x = 750 - (20*(i-20));
        (eval(foodClip))._y = 370;
    }
    else{
        (eval(foodClip))._x = 750 - (20*(i-40));
        (eval(foodClip))._y = 320;
    }

    //Make the clips drag and drop
    dragAndDrop(eval(foodClip));
}

//Function to remove the clips when the activity is finished
function removeFoodClips():Void{
    for (i = 0; i < question._value1; i++){
        var foodClip:String = "foodBuckets" + i;
        removeMovieClip(eval(foodClip));
    }
}

//Work out the number of elephants that need to be added
if (selectedAnimal._sex == "male"){
    //Just use the second value in the question
    var numberOfElephants:Number = question._value2;
}
else{
    //use the maximum multiple choice answer
    var maxAnswerAB:Number = Math.max(Number(answerA),
Number(answerB));
    var maxAnswerCD:Number = Math.max(Number(answerC),
Number(answerD));
    var maxAnswer:Number = Math.max(maxAnswerAB, maxAnswerCD);
    var numberOfElephants:Number = maxAnswer;
}

//Generate the appropriate number of elephants to match the question
for (i = 0; i < numberOfElephants; i++){
    //Dynamic variable name
    var elephantClip:String = "elephant" + i;

    //Attach, resize and position the clips on the stage
    this.attachMovie("ElephantSymbol", elephantClip, i);
    (eval(elephantClip))._width = 110;
    (eval(elephantClip))._height = 90;
    (eval(elephantClip))._x = 50 + 100*i;
    (eval(elephantClip))._y = 220;
}

```

```

        //Make the clips drag and drop
        dragAndDrop(eval(elephantClip));
    }

    //Function to remove the clips when the activity is finished
    function removeElephantClips():Void{
        for (i = 0; i < numberOfElephants; i++){
            var elephantClip:String = "elephant" + i;
            removeMovieClip(eval(elephantClip));
        }
    }
}

```

Layer	Actions	Keyframe	LionActivity
-------	---------	----------	--------------

```

stop();
currentPage = "LionActivity";
showMenu();

//menu button visibility
menu.instructions_btn._visible = false;
menu.help_btn._visible = false;
menu.question_btn._visible = true;
menu.map_btn._visible = true;
menu.zone_btn._visible = true;
menu.print_btn._visible = false;
menu.certificate_btn._visible = false;
menu.save_btn._visible = true;
menu.exit_btn._visible = true;

//Only show the selected zoo keeper
showSelectedKeeper();

//Initial display values
totalHutMoney = question._value1;
hutMoneyLeft = totalHutMoney;

hutMoney = "";
hutMoneyDisplay = 0;

//initial error message
errorMessage = "";

//Function to get the amount entered
function getAmountEntered():Void{
    hutMoney = Number(hutMoney);
    if (isNaN(hutMoney) == true){
        //If the value entered is not a number
        errorMessage = "Not a number!";
    }
}

```

```

        hutMoney = "";
        hutMoneyDisplay = 0;
        hutMoneyLeft = totalHutMoney;
    }
    else{
        hutMoneyDisplay = hutMoney;
        hutMoneyLeft = totalHutMoney - (hutMoney * 10);

        //Reset the text entry field
        hutMoney = "";
        enterHutMoney.text = "";
        errorMessage = "";
    }
}

//Enter value
enter_btn.onRelease = function(){
    getAmountEntered();
}

var keyListener:Object = new Object();

keyListener.onKeyDown = function() {
    if (Key.getCode() == Key.ENTER) {
        getAmountEntered();
    };
}

Key.addListener(keyListener);

hutMoneyKeyboard_btn.onRelease = function(){
    screenKeyboardEntry(_root.hutMoney, "hutMoney");
}

//menu button visibility
menu.instructions_btn._visible = false;
menu.help_btn._visible = false;
menu.question_btn._visible = true;
menu.map_btn._visible = true;
menu.zone_btn._visible = true;
menu.print_btn._visible = false;
menu.certificate_btn._visible = false;
menu.save_btn._visible = true;

```

Layer	Actions	Keyframe	LionActivity2
-------	---------	----------	---------------

```

stop();
currentPage = "LionActivity2";
showMenu();

//menu button visibility
menu.instructions_btn._visible = false;
menu.help_btn._visible = false;
menu.question_btn._visible = true;
menu.map_btn._visible = true;
menu.zone_btn._visible = true;
menu.print_btn._visible = false;
menu.certificate_btn._visible = false;
menu.save_btn._visible = true;

```

```

menu.exit_btn._visible = true;

//Only show the selected zoo keeper
showSelectedKeeper();

//Show the timeline of the length to match the hours in the question
timeLine4._visible = false;
timeLine8._visible = false;

//8 Hour timeline needed
if (question._value2 == 8){
    timeLine8._visible = true;
}

//4 Hour timeline needed
else if (question._value2 == 4){
    timeLine4._visible = true;
}

//Counter used to change the animal
var clipCounter:Number = 1;

//Generate the appropriate number of animal pointers to match the
question
for (i = 1; i <= question._value1; i++){
    //Dynamic variable name
    var animalClip:String = "animalPointer" + i;

    //Attach the clips, alternating between animals
    if(clipCounter == 1){
        this.attachMovie("GiraffePointer", animalClip, i);
    }
    else if (clipCounter == 2){
        this.attachMovie("ElephantPointer", animalClip, i);
    }
    else if (clipCounter == 3){
        this.attachMovie("ZebraPointer", animalClip, i);
    }
    else if (clipCounter == 4){
        this.attachMovie("LionPointer", animalClip, i);
    }

    //Set the position on the stage of the animal pointers
    (eval(animalClip))._x = 20 + (45*i);
    (eval(animalClip))._y = 450;

    //Make the animal pointers drag and drop
    dragAndDrop(eval(animalClip));
}

```

```

        clipCounter++;

        //Check if the counter needs to start from the beginning again
        if (clipCounter > 4){
            clipCounter = 1;
        }
    }

    //Function to remove the clips when the activity is finished
    function removeAnimalClips():Void{
        for (i = 1; i <= question._value1; i++){
            var animalClip:String = "animalPointer" + i;
            removeMovieClip(eval(animalClip));
        }
    }
}

```

Layer	Actions	Keyframe	QuestionRight
	<pre> stop(); currentPage = "QuestionRight"; showMenu(); //menu button visibility menu.instructions_btn._visible = true; menu.help_btn._visible = false; menu.question_btn._visible = false; menu.map_btn._visible = true; menu.zone_btn._visible = true; menu.print_btn._visible = false; menu.certificate_btn._visible = false; menu.save_btn._visible = true; menu.exit_btn._visible = true; //A question has been answered, therefore, the game state has changed gameSaved = false; //Display text, informing of which animal has been won winText = "Well done, you have won a " + selectedAnimal._sex + " " + selectedAnimal._animal + "!"; //Update the animal's won state to be true game._zoneStore[zoneIndex]._animalStore[animalIndex]._won = true; //Increment the number of animals won game._zoneStore[zoneIndex]._noAnimalsWon++; //Increment the zone and game points game._zoneStore[zoneIndex]._points = game._zoneStore[zoneIndex]._points + 10; </pre>		

```

game._gamePoints = game._gamePoints + 10;
_root.gamePoints = game._gamePoints;

```

```

//Only show the selected animal
showAnimal(selectedAnimal);

```

Layer	Actions	Keyframe	QuestionWrong
	<pre> stop(); currentPage = "QuestionWrong"; showMenu(); //menu button visibility menu.instructions_btn._visible = false; menu.help_btn._visible = true; menu.question_btn._visible = false; menu.map_btn._visible = true; menu.zone_btn._visible = true; menu.print_btn._visible = false; menu.certificate_btn._visible = false; menu.save_btn._visible = true; menu.exit_btn._visible = true; showSelectedKeeper(); //A question has been answered, therefore, game state has changed gameSaved = false; //Decrease zone and game score game._zoneStore[zoneIndex]._points = game._zoneStore[zoneIndex]._points - 5; game._gamePoints = game._gamePoints - 5; _root.gamePoints = game._gamePoints; //Increment the number of tries for that question game._zoneStore[zoneIndex]._animalStore[animalIndex]._questionStore[se lectedAnimal._questionNo]._noTries++; bubble1._visible = false; bubble2._visible = false; if(question._noTries == 1){ //The player can have one more try bubble1._visible = true; questionWrongHelp = "That was the wrong answer."; zooKeeperAdvice = "You can try that question one\nmore time.\n\n" </pre>		

```

        + "The Question Help might help
you work out the answer. "
        + question._helpText;
    //Hide new question button
    newQuestion_btn._visible = false;
}
else{
    //The player must now answer a different question
    bubble2._visible = true;
    questionWrongHelp = "That was the wrong answer again."
    zooKeeperAdvice = "\nYou will have to try a new\nquestion this
time.";

    //Hide try again and help activity buttons, show new question
button
    tryAgain_btn._visible = false;
    helpActivity_btn._visible = false;
    newQuestion_btn._visible = true;

    //Show menu button to get to new question rather than the help
activity
    menu.help_btn._visible = false;
    menu.question_btn._visible = true;

    //For the purpose of the prototype, set the number of tries for
the question just answered to 0 so it can be used again later if
necessary
    game._zoneStore[zoneIndex]._animalStore[animalIndex]._questions
tore[selectedAnimal._questionNo]._noTries = 0;

    if (selectedAnimal._questionNo + 1 ==
selectedAnimal._questionStore.length){
        //At the end of the question array, therefore, go back
to the first question

        game._zoneStore[zoneIndex]._animalStore[animalIndex]._questionN
o = 0;
    }
    else{
        //Set the new question to be the next question in the
question array

        game._zoneStore[zoneIndex]._animalStore[animalIndex]._questionN
o++;
    }
}

//Button actions

```

```

tryAgain_btn.onRelease = function(){
    gotoAndStop("Question");
}

helpActivity_btn.onRelease = function(){
    gotoAndStop(helpAct);
}

newQuestion_btn.onRelease = function(){
    gotoAndStop("Question");
}

```

Layer	Actions	Keyframe	Certificate
-------	---------	----------	-------------

```

stop();
currentPage = "Certificate";
showMenu();

//Menu button visibility
menu.instructions_btn._visible = true;
menu.help_btn._visible = false;
menu.question_btn._visible = false;
menu.map_btn._visible = true;
menu.zone_btn._visible = false;
menu.print_btn._visible = true;
menu.certificate_btn._visible = false;
menu.save_btn._visible = true;
menu.exit_btn._visible = true;

function deselectAnimals():Void{
    zebraSelected._visible = false;
    giraffeSelected._visible = false;
    elephantSelected._visible = false;
    lionSelected._visible = false;
}

function noShowAnimals():Void{
    finalCertificate.bwZebra._visible = false;
    finalCertificate.bwGiraffe._visible = false;
    finalCertificate.bwElephant._visible = false;
    finalCertificate.bwLion._visible = false;
}

//Select an animal
function selectAnimal(animalSelected):Void{
    //Hide all animal selection indicators
    deselectAnimals();
    //only show selected animal indicator
    animalSelected._visible = true;
}

```



```

}

//Show an animal on certificate
function showAnimal(animal):Void{
    //Hide all animals on certificate
    noShowAnimals();
    //only show selected animal on certificate
    animal._visible = true;
}

//Initially, no animal is selected or shown on certificate
deselectAnimals();
noShowAnimals();

function showSelectedAnimal(animalSelect, animalCertificate,
animalButton){
    animalButton.onRelease = function(){
        //Select and show zebra on certificate
        selectAnimal(animalSelect);
        showAnimal(animalCertificate);
    }
}

//Show the selected animal on certificate
showSelectedAnimal(zebraSelected, finalCertificate.bwZebra,
zebra_btn);
showSelectedAnimal(giraffeSelected, finalCertificate.bwGiraffe,
giraffe_btn);
showSelectedAnimal(elephantSelected, finalCertificate.bwElephant,
elephant_btn);
showSelectedAnimal(lionSelected, finalCertificate.bwLion, lion_btn);

//Date for certificate
var today_date:Date = new Date();
var date_str:String =
(today_date.getDate()+"-"+(today_date.getMonth()+1)+"-"+today_date.get
FullYear());

//Text for certificate
certText = "\n<b>" + game._playerName + ", Year " + game._schoolYear
+"</b>"
    + "\n\nZone: Super Safari"
    + "\nTopic: Division"
    + "\nDifficulty: " + game._difficulty
    + "\nScore: " + game._zoneStore[safari]._points + "
out of " + (game._zoneStore[safari]._noAnimals * 10)
    + "\n\nDate: " + date_str;

```

Layer	Actions	Keyframe	Instructions
	stop(); hideMenu();		
	//Button actions back_btn.onRelease = function(){ gotoAndStop(currentPage); }		
	next_btn.onRelease = function(){ gotoAndStop("Instructions2"); }		
Layer	Actions	Keyframe	Instructions2
	stop(); hideMenu();		
	//Button actions close_btn.onRelease = function(){ gotoAndStop(currentPage); }		
	back_btn.onRelease = function(){ gotoAndStop("Instructions"); }		